Tulun jiqi suanfa

图论
及其算法

殷剑宏　吴开亚　编著

Graph Theory and Its Algorithm

中国科学技术大学出版社

图论及其算法

殷剑宏 吴开亚 编著

中国科学技术大学出版社
2004·合肥

内 容 简 介

本书融有向图和无向图为一整体，系统地阐述了图论的基本概念、理论、方法及其算法。内容包括图的基本概念、Euler 图与 Hamilton 图、图论算法、树及其应用、平面图、独立集与匹配、网络流和 Petri 网。

书中附有大量例题和习题，而且大部分习题有详细解答。本书选材精炼全面，内容处理恰当且有新意，立论严谨，叙述条理清晰，语言流畅。

本书可用作高校计算机、电子、信息、管理、数学等专业本科生必修课教材，也可供相关专业的研究人员、教师及图论工作者参考。

图书在版编目(CIP)数据

图论及其算法/殷剑宏，吴开亚编著.—合肥：中国科学技术大学出版社，2003.7

ISBN 7-312-01558-1

Ⅰ.图…　Ⅱ.①殷…　②吴…　Ⅲ.①图论—高等学校—教材②图论算法—高等学校—教材　Ⅳ.O157.5

中国版本图书馆 CIP 数据核字(2003)第 022481 号

中国科学技术大学出版社出版发行
(安徽省合肥市金寨路 96 号，230026)
合肥学苑印务有限公司印制
合肥飞天图文艺术设计中心照排
全国新华书店经销

开本：850mm×1168mm/32　印张：9.25　字数：237 千
2003 年 7 月第 1 版　2004 年 1 月第 2 次印刷
印数：3001—6000 册
ISBN 7-312-01558-1/O·270　定价：18.00 元

前　言

图论是一门新兴学科，它发展迅速而又应用广泛。图论已广泛地应用于物理、化学、运筹学、计算机科学、电子学、信息论、控制论、网络理论、管理科学、社会科学等几乎所有学科领域。另一方面，随着这些学科的发展，特别是计算机科学的快速发展，又大大地促进了图论的发展。因此，国内外许多高等院校，都把图论列为数学、计算机、电子、信息、管理等专业的必修专业课程。

进入21世纪，国内越来越多的高校，为与国际接轨，拓展学生的知识面，增强学生竞争能力，增设了许多新的课程，同时又大大压缩了各门课程的教学学时，因此，在较少的学时内，如何加强学生对基本概念和基本理论的掌握，如何提高学生的实际应用能力，是我们高校教师又面临的一个新课题。正是针对这种实际情况，参考国内外许多优秀图论著作，结合作者的教学经验，对多年讲授的《图论及其算法》讲稿进行修改，形成了本书。在选材与内容编排等方面，不仅凝聚了作者多年的教学经验和教学体会，而且重点体现了既便于学生学习又利于教师教学的思想。

由于图论是一门较新的学科，所以大多数图论作者在其著作、论文、演讲中都习惯使用自己的一套术语和记号，而且许多图论著作只讨论有向图，或只讨论无向图，或将有向图与无向图分列讨论。本教材则把有向图与无向图融为一个整体，在介绍图论的基本概念、术语和结论时，参考国内外大多数作者的叙述，选择最为通俗且大众化的语言加以描述，且都以定义的形式加以规范，避免了概念的歧义性，为读者特别是初学者，勾画了清晰的图论轮廓，从而能轻松地进入图论的系统学习和研究。

本教材充分体现了图论在研究离散量的结构和相互间关系的独到之处。在内容安排上,各章之间既相互联系,具备教材的系统性和科学性,同时各章又相对独立,自成体系,给读者的学习提供了极大方便。另外,在介绍图论基本概念和基本理论的同时,还强调了图论算法,因为算法的研究是计算机科学的核心。因而既能为学生学习后继专业课程打下良好的数学基础,又能增强学生抽象思维能力,启迪学生研究算法的智慧,从而达到培养学生创造力的教学目标。

运用图论理论解决实际问题,有着非常独特与聪明之处。要形成图论思维,巧妙运用图论方法,除了深刻理解概念和理论,除了勤奋和激情,还需要智慧和技巧。为此,本教材每章都精选了适量例题和习题,书末附有习题解答,同学们在认真做题的过程中,将会惊叹图论方法的聪明和技巧,充分体会图论的美妙与魅力!

本教材贯彻了教育部有关图论课程教学大纲要求,能在大纲规定的教学学时内,达到图论课程的教学目标。而且,学习本教材,只需要具备线性代数及二元关系的初步知识。

作者

2003 年 4 月 28 日

目　次

第一章　图的基本概念

图论中所讨论的“图”，不是微积分、解析几何、几何学中讨论的图形，而是客观世界中某些具体事物间联系的一个数学抽象．如二元关系的关系图，在关系图中，我们不考虑点的位置及连线的长短曲直，而只关心哪些点之间有线相连．这种数学抽象就是“图”的概念．

第一节　图的概念

定义 1　图(graph)　所谓图 G 是一个三元组，记作 $G=\langle V(G),E(G),\varphi(G)\rangle$，其中

(1) $V(G)=\{v_1,v_2,\cdots,v_n\}$，$V(G)\neq\emptyset$，称为图 G 的**结点集合(vertex set)**．

(2) $E(G)=\{e_1,e_2,\cdots,e_m\}$ 是 G 的**边集合(edge set)**，其中 e_i 为 $\{v_j,v_t\}$ 或 $\langle v_j,v_t\rangle$．若 e_i 为 $\{v_j,v_t\}$，称 e_i 为以 v_j 和 v_t 为**端点(end vertices)**的**无向边(undirected edge)**；若 e_i 为 $\langle v_j,v_t\rangle$，称 e_i 为以 v_j 为**起点(origin)**，v_t 为**终点(terminus)**的**有向边(directed edge)**．

(3) $\varphi(G):E\to V\times V$ 称为**关联函数(incidence function)**．

例 1　已知图 $G=\langle V(G),E(G),\varphi(G)\rangle$，其中

$V(G)=\{v_1,v_2,v_3,v_4,v_5,v_6\}$

$E(G)=\{e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8,e_9,e_{10}\}$

$\varphi(G):E\to V\times V$，且

$\varphi(e_1)=\langle v_1,v_2\rangle$　　$\varphi(e_2)=\langle v_3,v_2\rangle$　　$\varphi(e_3)=\{v_3,v_3\}$

$\varphi(e_4)=\langle v_4,v_3\rangle$　　$\varphi(e_5)=\langle v_4,v_2\rangle$　　$\varphi(e_6)=\langle v_4,v_2\rangle$

$\varphi(e_7)=\langle v_5,v_2\rangle$　　$\varphi(e_8)=\langle v_2,v_5\rangle$　　$\varphi(e_9)=\{v_3,v_5\}$

$\varphi(e_{10})=\{v_3,v_5\}$

图 G 的一个图形表示(diagrammatic presentation)如图 1-1.

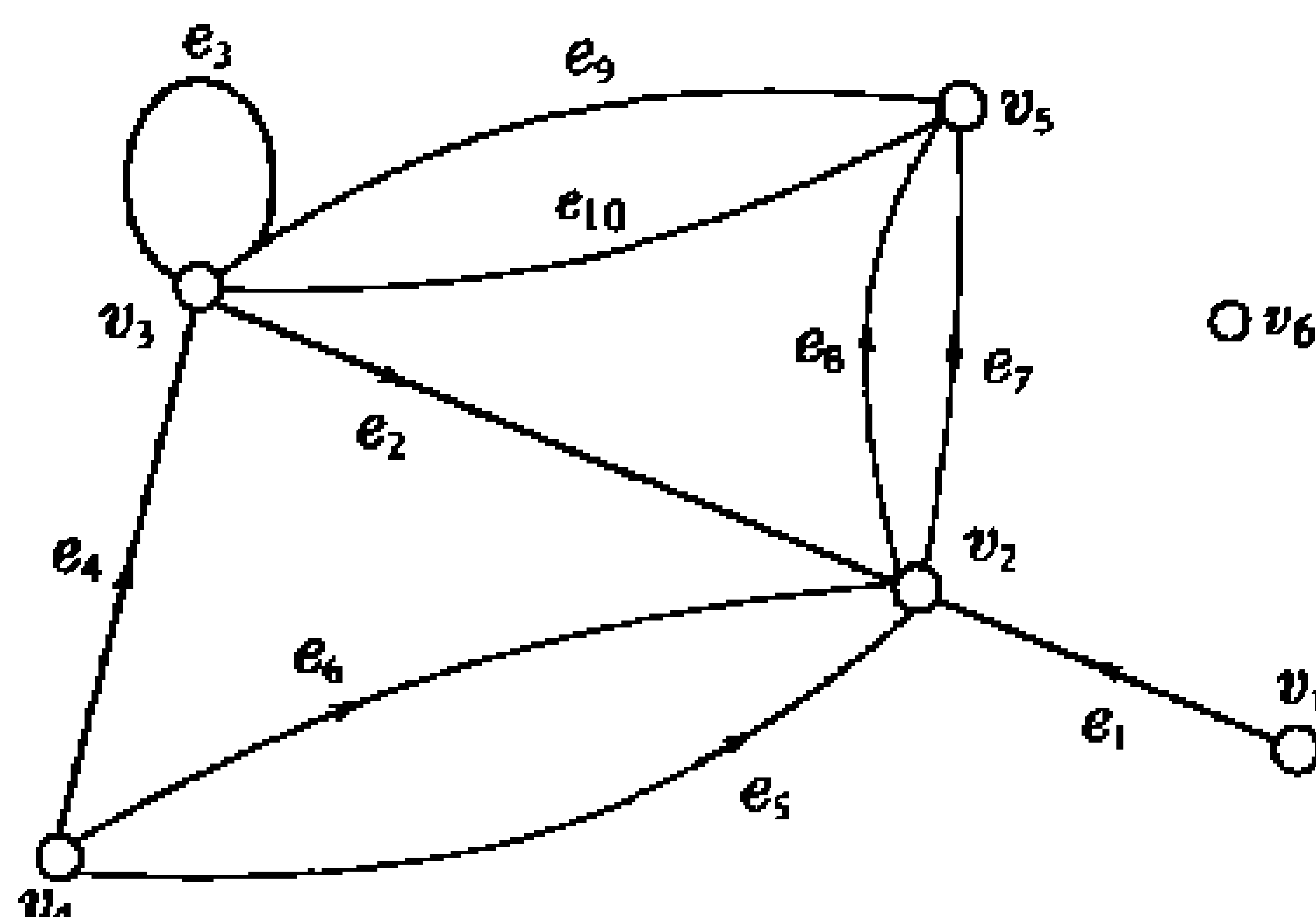


图 1-1

图 G 的图形表示,即用平面上的小圆圈表示图 G 的顶点,用点与点之间的连线表示图 G 中的边.图的图形表示使得抽象定义的图具有直观性,有助于我们进行思考和理解图的性质.当然,同一个图 G 可以有许多形状不同的图形表示.

定义 2　邻接结点(adjacent vertices)　关联于同一条边的两个结点称为邻接结点.

孤立结点(isolated vertex)　不与任何结点相连接的结点称为孤立结点(度数为零的结点).

邻接边(adjacent sides)　关联同一个结点的两条边称为邻接边.

环(loop)　两端点相同的边称为环或自回路(circuit).

平行边(parallel edges)　两个结点间方向相同的若干条边称为平行边或重边(multiple edges).

对称边(symmetric edges) 两端点相同但方向互为相反的两条有向边称为对称边.

定义 3 无向图(undirected graph) 每条边都是无向边的图称为无向图.

有向图(directed graph) 每一条边都是有向边的图称为有向图.

混合图(mixed graph) 图中不全是有向边,也不全是无向边的图称为混合图.

零图(null graph) 仅有一些孤立结点的图称为零图或空图(**empty graph**).

平凡图(trivial graph) 只有一个孤立结点的图称为平凡图.

定义 4 多重图(multigraph) 含有平行边的图称为多重图.

简单图(simple graph) 无环并且无平行边的图称为简单图.

完全图(complete graph) 任何不同两结点之间都有边相连的简单无向图称为完全图.

注释 1 (1)在简单图 $G=\langle V(G),E(G),\varphi(G)\rangle$中,以 x 为起点 y 为终点的边至多有一条,因此,图中的边可以直接用顶点对表示,而关联函数 φ 就可以直接表示在其边集中,故可简记为 $G=\langle V(G),E(G)\rangle$.

(2)对于无向图 G,将 G 中的每条边用两条与 e 有相同端点的对称边 e 和 e' 来代替后得到一个有向图 D,这样得到的有向图 D 称为 G 的**对称有向图(symmetric digraph)**.由此可见,无向图可视为特殊的有向图.

(3)n 个结点的完全图记作 K_n,完全图 K_n 有 $C_n^2=\frac{1}{2}n(n-1)$ 条边.完全图的对称有向图称为**完全有向图(complete digraph)**,记作 K_n^*.

(4)图 G 的顶点个数还称为图 G 的**阶(order)**.

(5)对于有向图 D,去掉边上的方向得到的无向图 G 称为 D 的**基础图**(**underlying graph**).反之,任一个无向图 G,将 G 的边指定一个方向得到一个有向图 D,称 D 为 G 的一个**定向图**(**oriented graph**).

例 2 证明:在任意六个人的聚会上,要么有三人曾相识,要么有三人不曾相识.

证明 我们用 A、B、C、D、E、F 代表这六个人,若二人曾相识,则代表这二人的两顶点间连一条红边;否则连一条蓝边.于是原来的问题就等价于证明这样得到的图中必含有同色三角形.考察某一顶点,设为 F.与 F 关联的边中必有三条同色,不妨设它们是三条红边 FA、FB、FC.再看三角形 ABC.如果它有一条红边,设为 AB,则 FAB 是红边三角形;如果三角形 ABC 没有红边,则它本身就是蓝边三角形.

第二节 图的顶点度和图的同构

定义 1 设 G 是任意图,x 为 G 的任一结点,与结点 x 关联的边数(一条环要计算两次)称为 x 的**度数**(**degree**).记作 $\deg(x)$ 或 $d(x)$.

设 D 是任意有向图,x 为 G 的任一结点,射入 x 的边数称为 x 的**入度**(**in-degree**),记作 $\deg^{+}(x)$ 或 $d^{+}(x)$;射出 x 的边数称为 x 的**出度**(**out-degree**),记作 $\deg^{-}(x)$ 或 $d^{-}(x)$.

注释 1 (1)有向图 D 中,任意结点 x 的度数 $\deg(x)=\deg^{+}(x)+\deg^{-}(x)$.

(2)用 $\Delta(G)$ 和 $\delta(G)$ 分别表示 G 的最大顶点度和最小顶点度,即 $\Delta(G)=\max\{d_G(x)\mid x\in V(G)\}$,$\delta(G)=\min\{d_G(x)\mid x\in V(G)\}$.

(3)设 D 是有向图,记

$$\Delta^+(D)=\max\{d_D^+(x)\mid x\in V(D)\},$$
$$\Delta^-(D)=\max\{d_D^-(x)\mid x\in V(D)\},$$
$$\delta^+(D)=\min\{d_D^+(x)\mid x\in V(D)\},$$
$$\delta^-(D)=\min\{d_D^-(x)\mid x\in V(D)\}.$$

(4)已知图 $G=\langle V,E\rangle$，$S\subseteq V$，$S\neq\emptyset$，$\forall v\in V$，称 $N_s(v)=\{u\mid u\in S$，且 u 与 v 相邻$\}$为 v 在 S 中的**邻域**. 特别，$N_G(v)$常简记为 $N(v)$. 显然，当 G 是简单图时，$d(v)=|N(v)|$.

定义 2 设 G 为无向图，对于 G 的每个结点 x，若 $d(x)=K$，则称 G 为 **K 正则的(k-regular)无向图**.

设 D 为有向图，对于 G 的每个结点 x，若 $d^+(x)=d^-(x)$，则称 D 为**平衡有向图(balanced digraph)**. 其中 x 称为**平衡点(balanced vertex)**.

在有向图 D 中，若 $\Delta^+(D)=\Delta^-(D)=\delta^+(D)=\delta^-(D)=K$，则称 D 为 **K 正则有向图**.

定理 1 每个图中，结点度数的总和等于边数的二倍. 即 $\sum\limits_{x\in V}\deg(x)=2|E|$.

证明 设图 G 有 m 条边，每条边均连接两个结点，即给两个结点的度数各加 1，m 条边总共提供 $2m$ 度数，即 $\sum\limits_{x\in V}\deg(x)=2m$.

定理 2 每个图中，度数为奇数的结点必定是偶数个.

证明 设 V_1，V_2 分别是 G 中奇数度数和偶数度数的结点集，则由定理 1 知，

$$\sum_{x\in V_1}\deg(x)+\sum_{x\in V_2}\deg(x)=\sum_{x\in V}\deg(x)=2|E|$$

由于 $\sum\limits_{x\in V_2}\deg(x)$ 为偶之和，必为偶数，而 $2|E|$ 是偶数，故得

$\sum_{x\in V_1}\deg(x)$为偶数，即$|V_1|$是偶数.

定理 3 在任何有向图中，所有结点的入度之和等于所有结点出度之和.

证明 因为每条有向边必对应一个入度和一个出度，若一个结点具有一个入度或出度，则必关联一条有向边，因此，有向图中各结点的入度之和等于边数，各结点出度之和也等于边数.故定理得证.

例如，图 1-1 中，$d(v_1)=1, d(v_2)=6, d(v_3)=6, d(v_4)=3, d(v_5)=4, d(v_6)=0. \delta(G)=0, \Delta(G)=6$.

如果 $V(G)=\{v_1, v_2, \cdots, v_p\}$，称非负整数序列$(d(v_1), d(v_2), \cdots, d(v_p))$为图 G 的**度序列**.

例如，图 1-1 所示图 G 的度序列为(1,6,6,3,4,0).

推论 1 非负整数序列$(d_1, d_2, \cdots, d_p)$是某个图的度序列当且仅当 $\sum_{i=1}^{p} d_i$ 是偶数.

证明 由定理 1 可知必要性成立.对于充分性，取 p 个相异顶点 $v_1, v_2, \cdots, v_p$，若 d_i 是偶数，就在 v_i 处作 $d_i/2$ 个环；若 d_i 是奇数，在 v_i 处作$(d_i-1)/2$ 个环.由于 $\sum_{i=1}^{p} d_i$ 为偶数，故 $d_1, d_2, \cdots, d_p$ 中有偶数个奇数顶点，从而将所有与奇数 d_i 相对应的这些顶点 v_i 两两配对并连上一条边.最后所得的度序列就是$(d_1, d_2, \cdots, d_p)$.证毕

值得注意的是，以非负整数序列$(d_1, d_2, \cdots, d_p)$($\sum_{i=1}^{p} d_i$ 是偶数)为度序列的图一般有很多.

例如，图 1-2 所示的 G_1 和 G_2 的度序列均是(7,3,1,4,6,5).

简单图的度序列称为**图序列**，图序列的讨论或判断要比度序

列的讨论困难得多.即使知道非负整数序列$(d_1, d_2, \cdots, d_p)$是图序列,要构造相应的简单图仍是困难的.

Erdös 和 Callai 在 1960 年给出了图序列的一个判别方法.

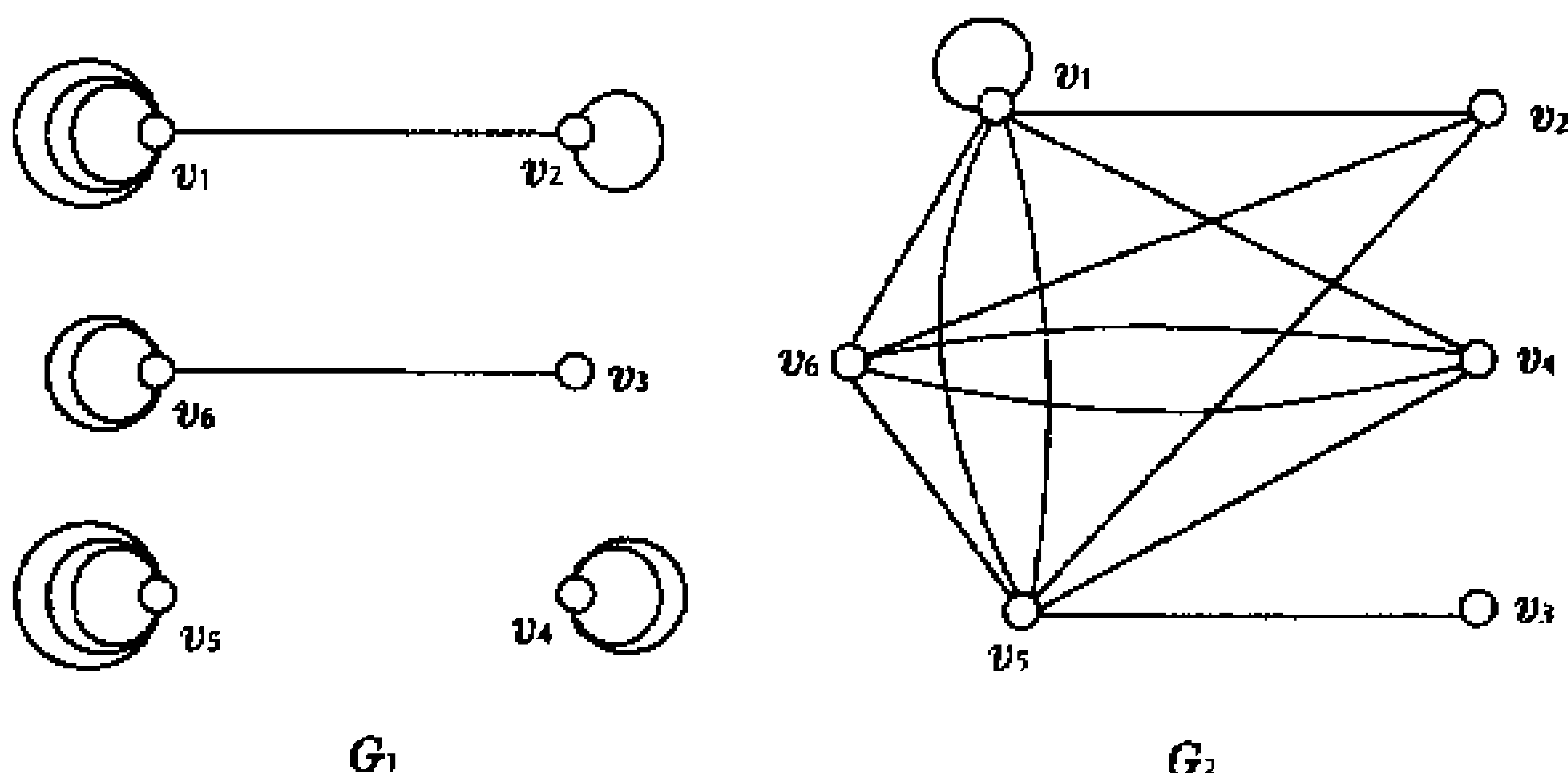


图 1-2

定理 4 非负整数序列$(d_1, d_2, \cdots, d_p)$$(d_1 \geqslant d_2 \geqslant \cdots \geqslant d_p)$是图序列当且仅当$\sum_{i=1}^{p} d_i$是偶数,并且对一切整数$k$,$1 \leqslant k \leqslant p-1$,有

$$\sum_{i=1}^{k} d_i \leqslant k(k-1) + \sum_{i=k+1}^{p} \min\{k, d_i\}$$

证明略.

定义 3 设$G_1 = \langle V_1, E_1 \rangle$和$G_2 = \langle V_2, E_2 \rangle$是简单图,若存在一个从$V_1$到$V_2$的双射函数$f$,且$f$具有这样的性质,对$V_1$中的所有$x$和$y$,$x$和$y$在$G_1$中相邻,当且仅当$f(x)$和$f(y)$在$G_2$中相邻,就说$G_1$与$G_2$是**同构的**(**isomorphic**),记作$G_1 \cong G_2$.这样的函数f称为**同构函数**.

换句话说,当两个简单图同构时,两个图的顶点之间具有保持相邻关系的一一对应.

到目前为止，判断两个图是否同构，还只能根据定义．也就是说，两个图是否同构还没有很简便的判别方法．

两图同构的必要条件是：(1)两图结点数相等．(2)边数相等．(3)度数相同的结点个数相等．但这仅是必要条件，不是充分条件．

例1 如图1-3中，$G_1 \cong G_2$，$G_3 \cong G_4$，$G_5 \cong G_6$，$G_7 \cong G_8 \cong G_9$．

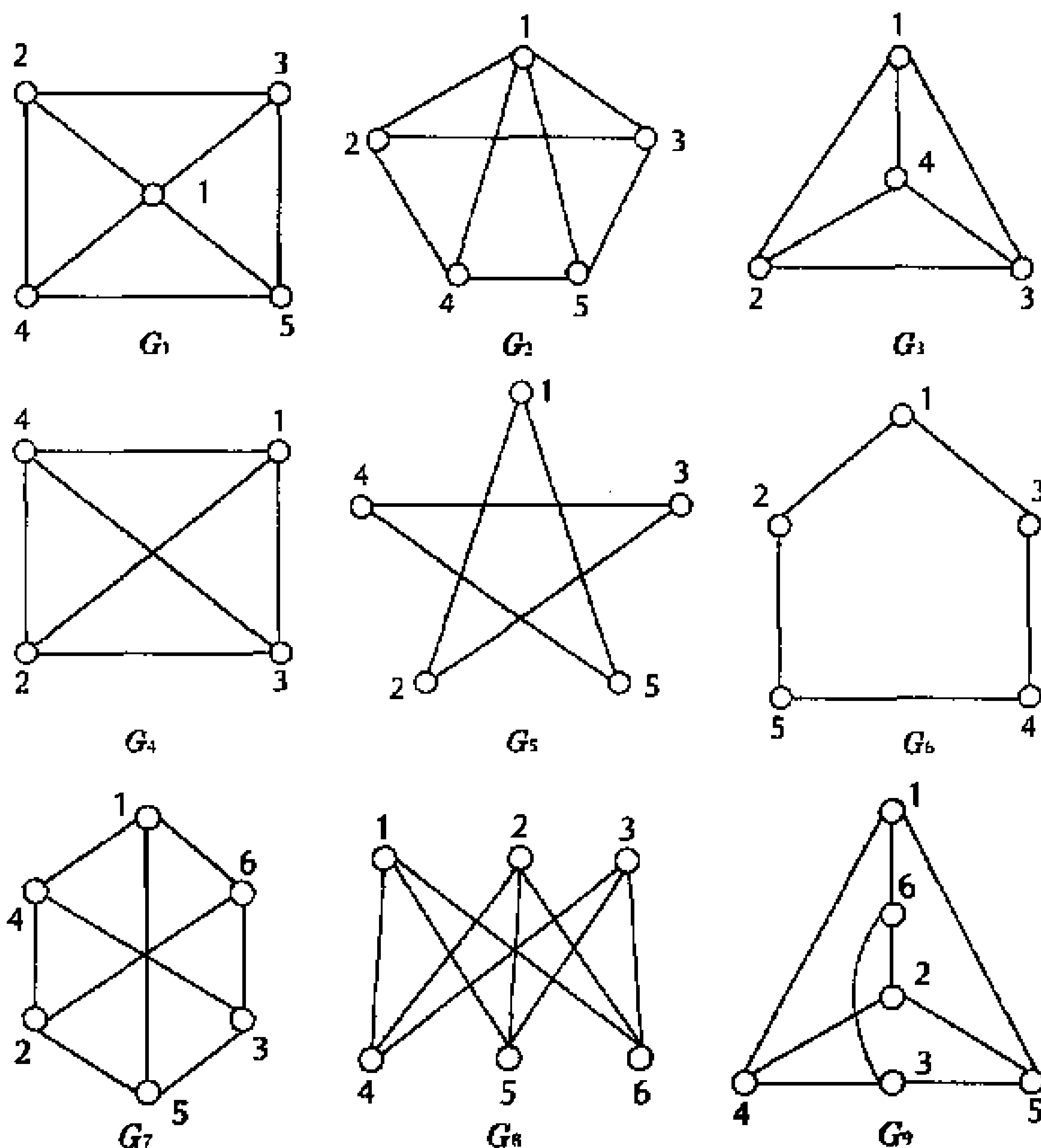


图1-3

对于两个同构的图，易见它们有相同的结构，差异只是顶点和边的名称不同，或两个图的形状不同，由于我们主要关注的是图的结构性质，所以在画图时常常可以省略顶点和边的标号．一个无标号图就认为是同构图的等价类的代表．

例 2　画出所有不同构的具有 5 个结点，3 条边的简单无向图．

解　共有 4 个，分别如图 1-4 中的 G_1,G_2,G_3,G_4．

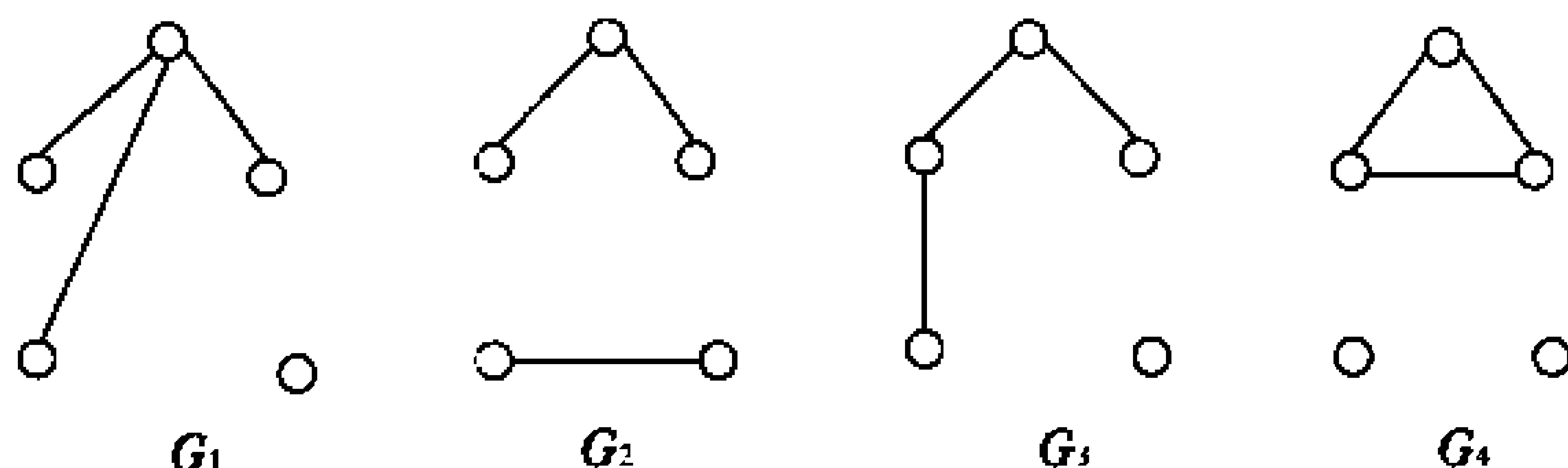


图 1-4

例 3　是否可以画一个简单无向图，使各点度数与下列的序列一致．

(1)2,2,2,2,2,2

(2)2,2,3,4,5,6

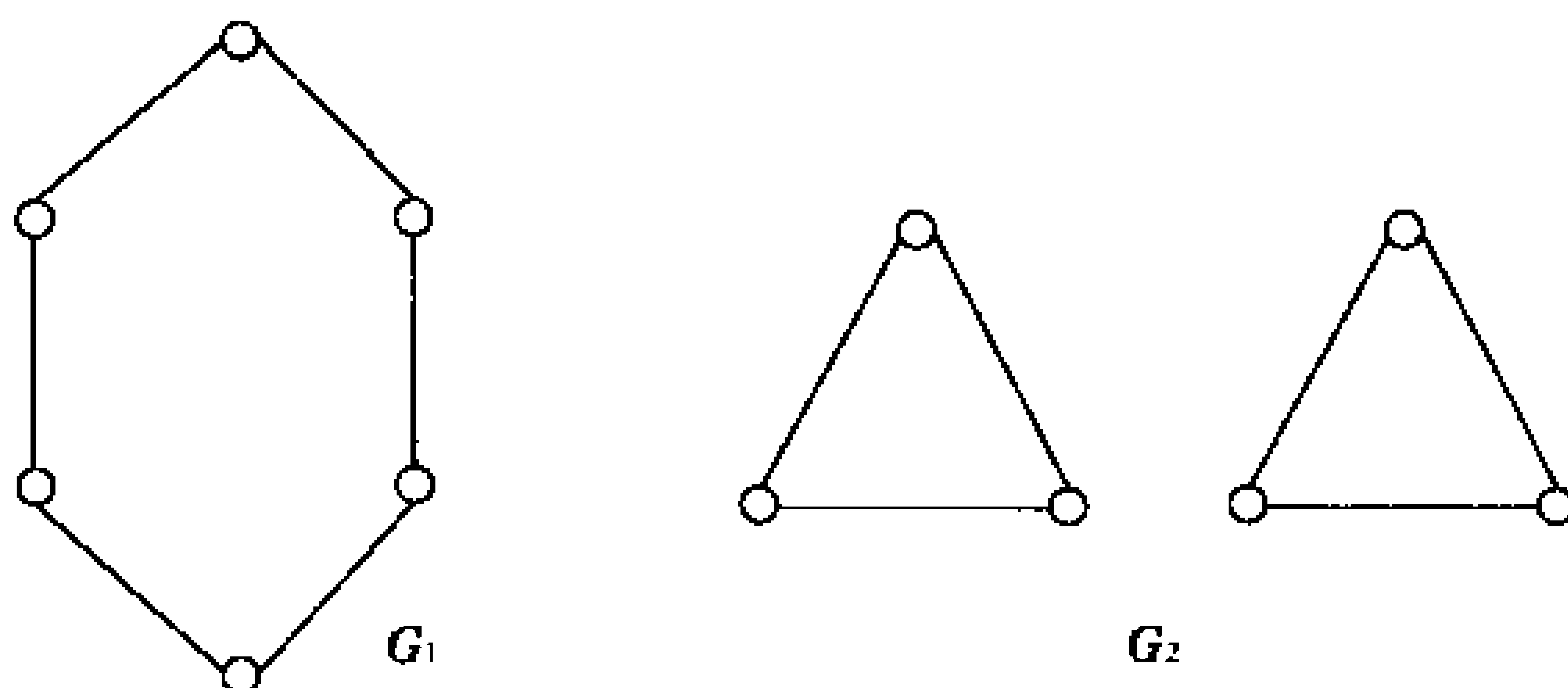


图 1-5

(3)1,2,3,4,4,5

解 (1)可以.如图 1-5 的 G_1 或 G_2.

(2)不可以.在 6 个结点的简单无向图中，其中每个结点最多与其余 5 个结点相邻,即 $\Delta(G)\leqslant 5$.

(3)不可以.任何图中,度数为奇数的结点只能有偶数个.

第三节 图的运算

定义 1 设 $G=\langle V,E,\varphi\rangle$与 $G_1=\langle V_1,E_1,\varphi_1\rangle$是任两个图.若 $V_1\subseteq V,E_1\subseteq E$,且 φ_1 是 φ 在E_1 上的限制,则称 G_1 是 G 的**子图(subgraph)**,记作 $G_1\subseteq G$.称 G 为G_1 的**母图(supergraph)**.

若 $G_1\subseteq G$,且 $G_1\neq G$(即 $V_1\subset V$ 或$E_1\subset E$),则称 G_1 是 G 的**真子图(prope subgraph)**.

若 $G_1\subseteq G$,且 $V_1=V$,则称 G_1 是 G 的**生成子图或支撑子图(spanning subgraph)**.

设 $V_1\subseteq V$,且 $V_1\neq\emptyset$,以 V_1 为顶点,以两端点均在 V_1 中的全体边为边集的 G 的子图,称为 V_1 的**导出子图(induced subgraph)**,记作 $G[V_1]$.

设 $E_1\subseteq E$,且 $E_1\neq\emptyset$,以 E_1 为边集,以 E_1 中的边关联的全部顶点为顶点集的 G 的子图,称为 E_1 的导出子图,记作 $G[E_1]$.

特别,若 $V_1\subset V$,且 $V_1\neq\emptyset$,则以 $G-V_1$ 表示从 G 中删去 V_1 内的所有点以及与这些顶点相关联的边所得到的子图.若 $V_1=\{v\}$,常把 $G-\{v\}$简记为 $G-v$.类似地,设 $E_1\subset E$,且 $E_1\neq\emptyset$,$G-E_1$ 表示在 G 中删去E_1 中所有边所得的子图.同样,$G-\{e\}$简记为 $G-e$.

定义 2 设 $G=\langle V,E\rangle$是 n 阶无向简单图.以 V 为顶点集,以所有能使 G 成为完全图K_n 的添加边组成的集合为边集的图,

称为 G 相对于完全图 K_n 的补图，简称 G 的补图(complement graph)，记作 $\bar{G}$.

例如，图 1-6(b)所示的图为图 1-6(a)所示的图的补图.

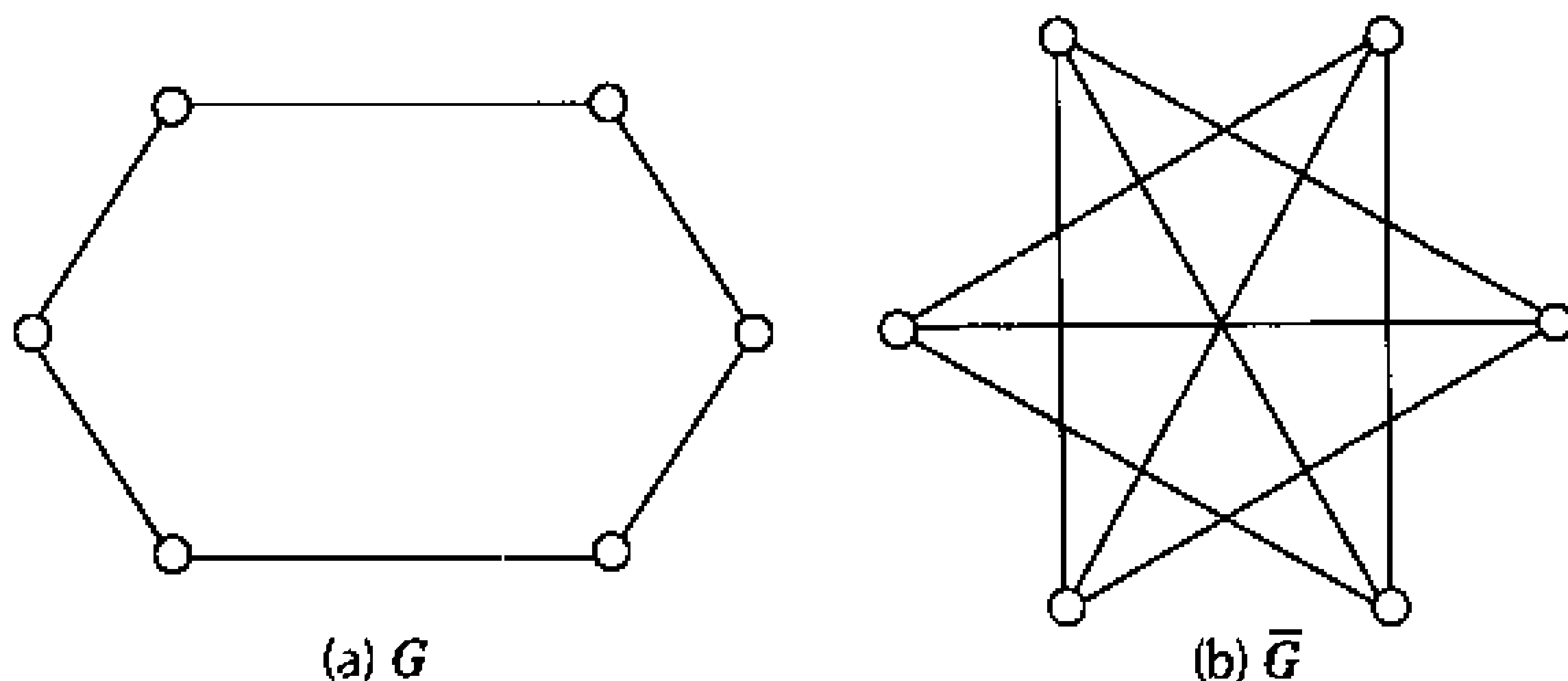


图 1-6

定义 3 设 G_1 和 G_2 都是图 G 的子图.

G_1 和 G_2 的**并**(union)，记作 $G_1 \cup G_2$：仅由 G_1 和 G_2 中所有边组成的图.

G_1 和 G_2 的**交**(cap)，记作 $G_1 \cap G_2$：仅由 G_1 和 G_2 的公共边组成的图.

G_1 和 G_2 的**差**(difference)，记作 $G_1 - G_2$：仅由 G_1 中去掉 G_2 中的边组成的图.

G_1 和 G_2 的**环和**(ring sum)，记作 $G_1 \oplus G_2$：在 G_1 和 G_2 的并中去掉 G_1 和 G_2 的交所得到的图，即

$$G_1 \oplus G_2 = (G_1 \cup G_2) - (G_1 \cap G_2) = (G_1 - G_2) \cup (G_2 - G_1)$$

例如，设 G_1 和 G_2 为图 1-7 所示，它们的并、交、差、环和分别如图 1-8 所示.

定义 4 设 G_1 和 G_2 是任两个无向图，G_1 和 G_2 的**笛卡儿积**(Cartesian products)为图 $G = G_1 \times G_2$，其中图 G 满足：$V(G) = V(G_1) \times V(G_2)$；$G$ 中的两个顶点 $\langle a, b\rangle$ 和 $\langle c, d\rangle$ 是邻接的当且

仅当 $a=c$ 且 $\{b,d\}\in E(G_2)$ 或者 $b=d$ 且 $\{a,c\}\in E(G_1)$.

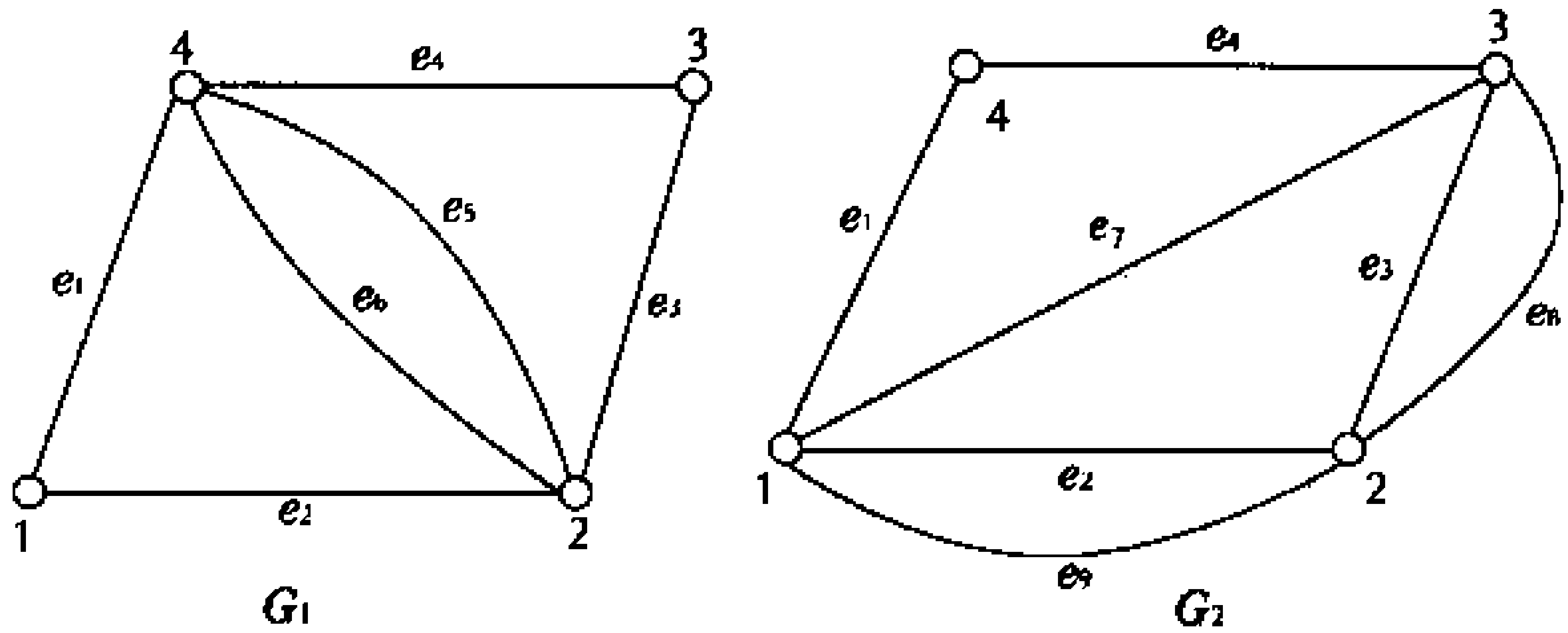


图 1-7

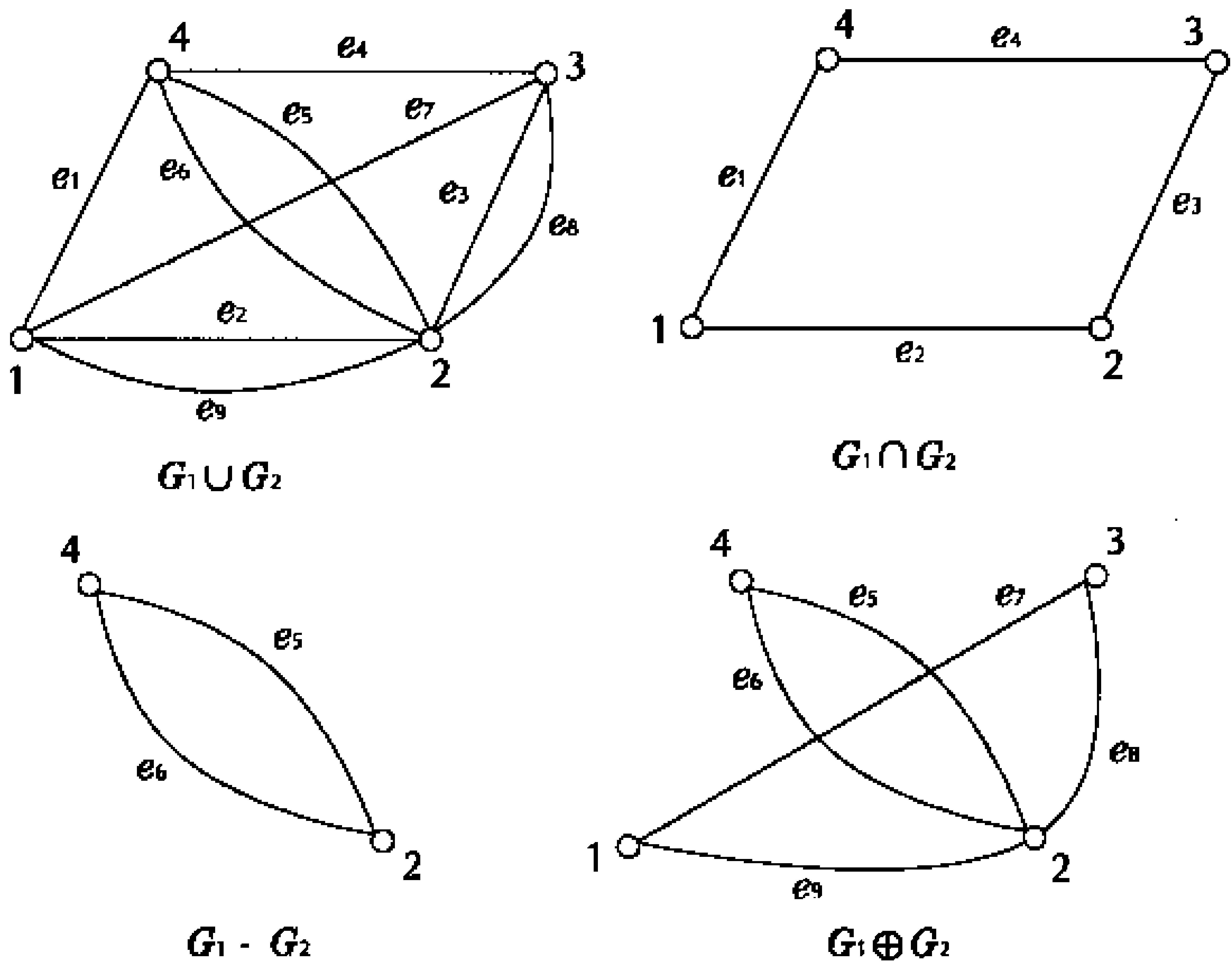


图 1-8

例如,图 1-9 所示分别为图 $G_1,G_2,G_1\times G_2$.

通过图的笛卡儿积运算,可归纳地定义一个重要的图类——

n 立方体 $Q_n(n \geqslant 1)$. 当 $n=1, Q_n=K_2$; 当 $n>1, Q_n=Q_{n-1} \times K_2$. 例如, 图 1-10 所示分别为图 Q_1, Q_2, Q_3, Q_4.

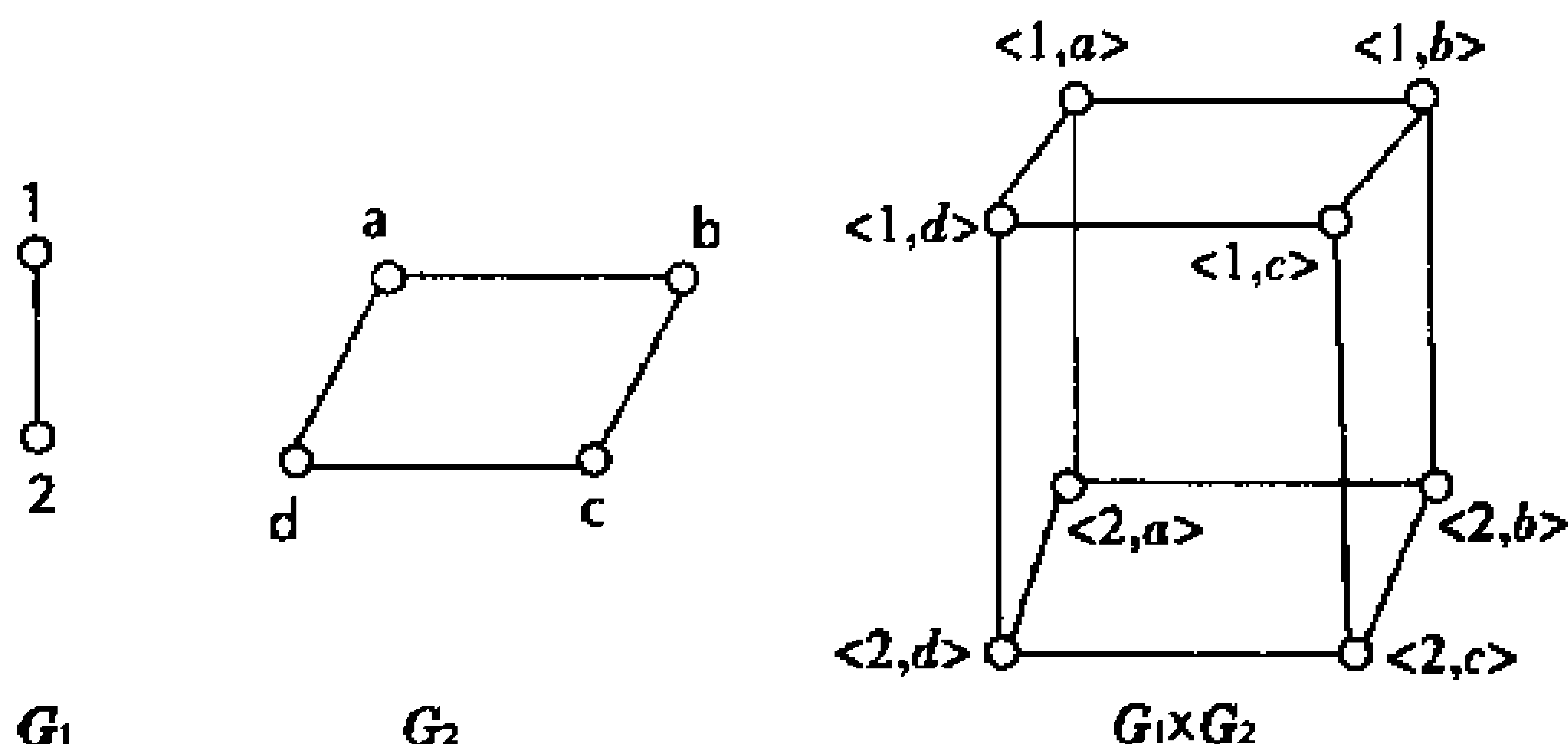


图 1-9

n 立方体 Q_n 也可以看作是用顶点表示 2^n 个长度为 n 的位串的图. 两个顶点相邻, 当且仅当它们所表示的位串恰恰差一位.

第四节　路与连通图

定义 1　设 u 和 v 是任意图 G 的顶点, 图 G 的一条 $u-v$ **链**(**chain** 或 **walk**)是有限的顶点和边交替序列 $u_0e_1u_1e_2\cdots u_{n-1}e_nu_n$ $(u=u_0, v=u_n)$, 其中与边 $e_i(1 \leqslant i \leqslant n)$ 相邻的两顶点 u_{i-1} 和 u_i 正好是 e_i 的两个端点. 数 n (链中出现的边数)称为链的**长度**(**length**). $u(u_0)$ 和 $v(u_n)$ 称为链的**端点**(**end-vertices**), 其余的顶点称为链的**内部点**(**internal vertices**). 一条 $u-v$ 链, 当 $u \neq v$ 时, 称它为**开的**, 否则称为**闭的**. 边互不同的链称为**迹**(**trail**), 内部点互不同的链称为**路**(**path**).

注释 1　(1)在一条链中, 顶点和边可以重复.

(2)若 G 是简单图, G 中的链 $u_0e_1u_1e_2\cdots u_{n-1}e_nu_n$ 还可用结点序列 $u_0u_1\cdots u_{n-1}u_n$ 表示.

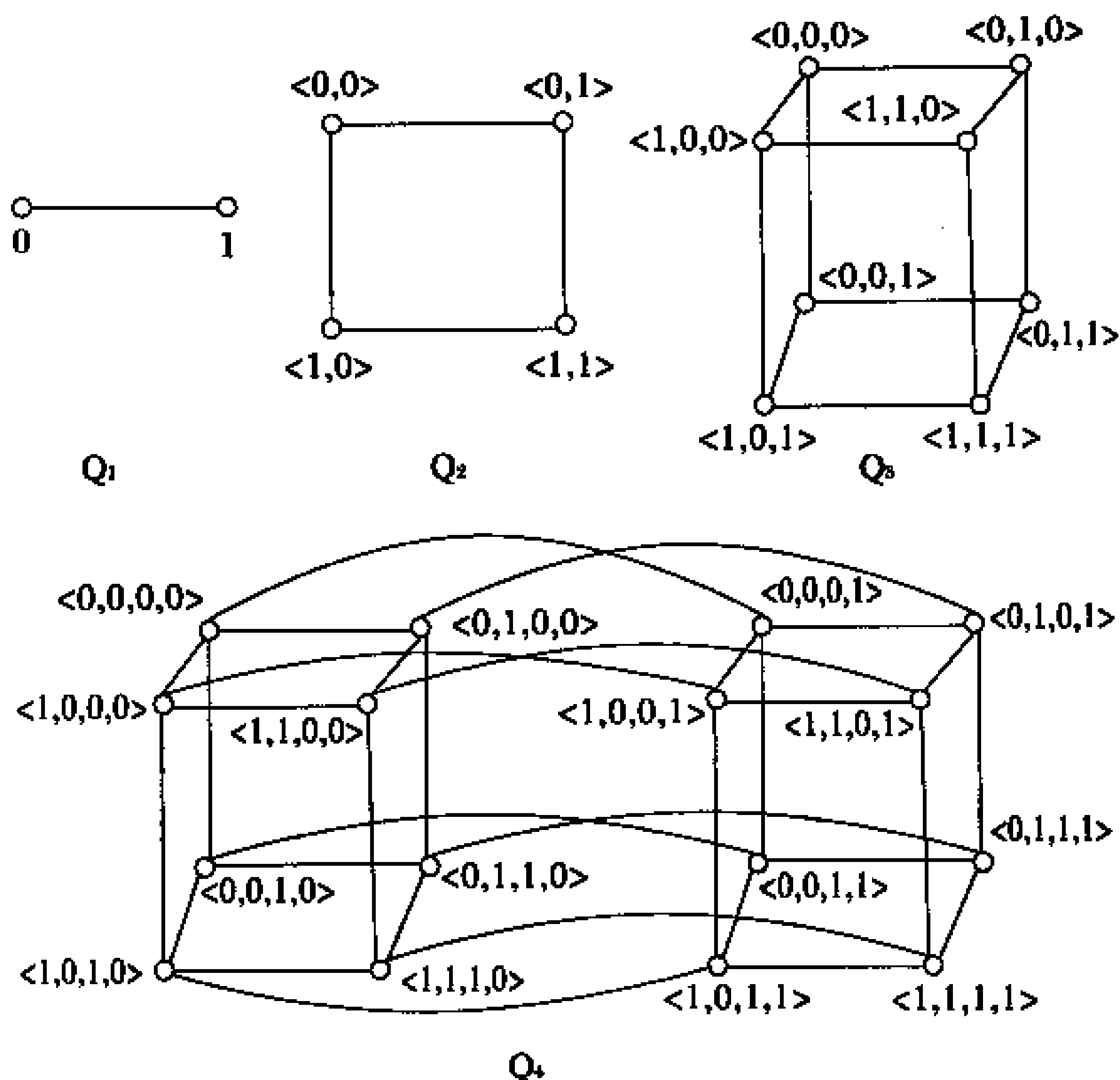


图 1-10

(3)不含边(即长度为 0)的链称为**平凡链**.

(4)设 W 是有向图 D 中 $u-v$ 链(迹、路),指定 W 的方向从 u 到 v.若 W 中所有边的方向与此方向一致,则称 W 为 D 中从 u 到 v 的**有向链(迹、路)**.

下面给出图 1-11 所示图 G 的链(迹、路).

x_1-x_3 链:$x_1e_9x_2e_8x_5e_2x_5e_3x_4e_3x_5e_6x_3$

x_1-x_3 迹:$x_1e_9x_2e_8x_5e_2x_5e_3x_4e_4x_3$

x_1-x_3 路:$x_1e_9x_2e_8x_5e_3x_4e_4x_3$

有向 x_1-x_3 链:$x_1e_1x_5e_2x_5e_6x_3e_4x_4e_5x_3$

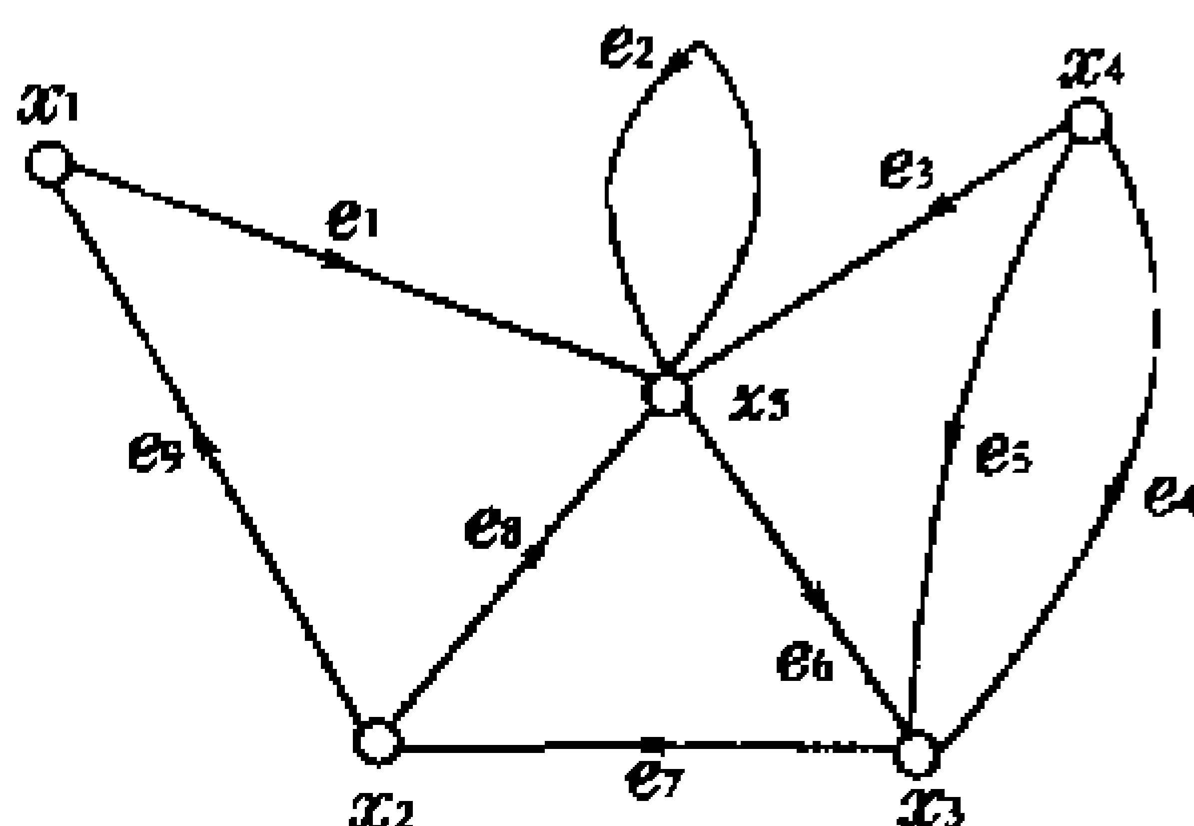


图 1-11

有向 x_1-x_3 迹：$x_1e_1x_5e_2x_5e_6x_3$

有向 x_1-x_3 路：$x_1e_1x_5e_6x_3$

定义 2 两端点相同的迹(即闭迹)称为回(circuit). 两端点相同的路(即闭路)称为圈(cycle)或回路(circuit). 长度为 K、奇数、偶数的回(圈)分别称为 K、奇(odd)、偶(even)回(圈). 有向闭迹(闭路)称为有向回(有向圈).

定理 1 若简单图 G 中每个顶点的度数至少是 $k(k\geqslant2)$，则 G 中必然含有一个长度至少是 $k+1$ 的圈.

证明 在 G 的所有路中，取一条长度最长的路 p，记 $p=v_0v_1\cdots v_{t-1}v_t$. 则 v_0 的所有邻接点全在 p 中，由于 $d(v_0)\geqslant k\geqslant 2$，所以 v_0 至少有 k 个邻接点，设所有邻接点为 $v_{i_1},v_{i_2},\cdots,v_{i_s}$，$1\leqslant i_1<i_2<\cdots<i_s\leqslant t$，其中 $s=d(v_0)\geqslant k\geqslant 2$，则

$c=v_0v_1v_2\cdots v_{i_s}v_0$ 就是 G 的一个长为 i_s+1 的圈，显然 $i_s+1\geqslant k+1$.

对于有向图，类似地可以证明.

定理 2 设简单图 G 中每个顶点的度数至少是 3，则 G 含有偶圈.

证明 设 $p=v_0v_1\cdots v_t$ 是 G 的一条最长路，则 v_0 的所有邻接点全在 p 内. 设 v_0 的全部邻接点为 $v_{i_1},v_{i_2},\cdots,v_{i_s}$ $(1\leqslant i_1<i_2$

$<\cdots<i_s\leqslant t)$,其中 $s=d(v_0)\geqslant 3$,在 G 中取三个圈

$$c_1=v_0v_1\cdots v_{i_2}v_0$$

$$c_2=v_0v_1\cdots v_{i_s}v_0$$

$$c_3=v_0v_{i_2}v_{i_2+1}\cdots v_{i_s}v_0$$

它们的长度分别为 i_2+1,i_s+1 和 i_s-i_2+2.这三个数中至少有一个是偶数.即 c_1,c_2 和 c_3 中至少有一个是偶圈.

定义 3　给定无向图 $G=\langle V(G),E(G),\varphi(G)\rangle$,$x,y\in V(G)$,若图 G 中存在连接 x 和 y 的路,称结点 x 和 y 是**连通的**(**connected**).规定 x 到自身总是连通的.

容易验证,结点集 $V(G)$上的顶点间的连通关系是 $V(G)$上的等价关系,该等价关系确定 $V(G)$的一个划分$\{V_1,V_2,\cdots,V_m\}$,使得当且仅当两个顶点 x 和 y 属于同一子集 V_i 时,x 和 y 才是连通的.V_i 在 G 中的导出子图 $G[V_1],G[V_2],\cdots,G[V_m]$ 称为 G 的**连通分支或分支**(**component**),m 称为 G 的**连通分支数**(**number of components**),记作 $W(G)=m$.

例如,图 1-12 所示图 G 有 4 个连通分支.

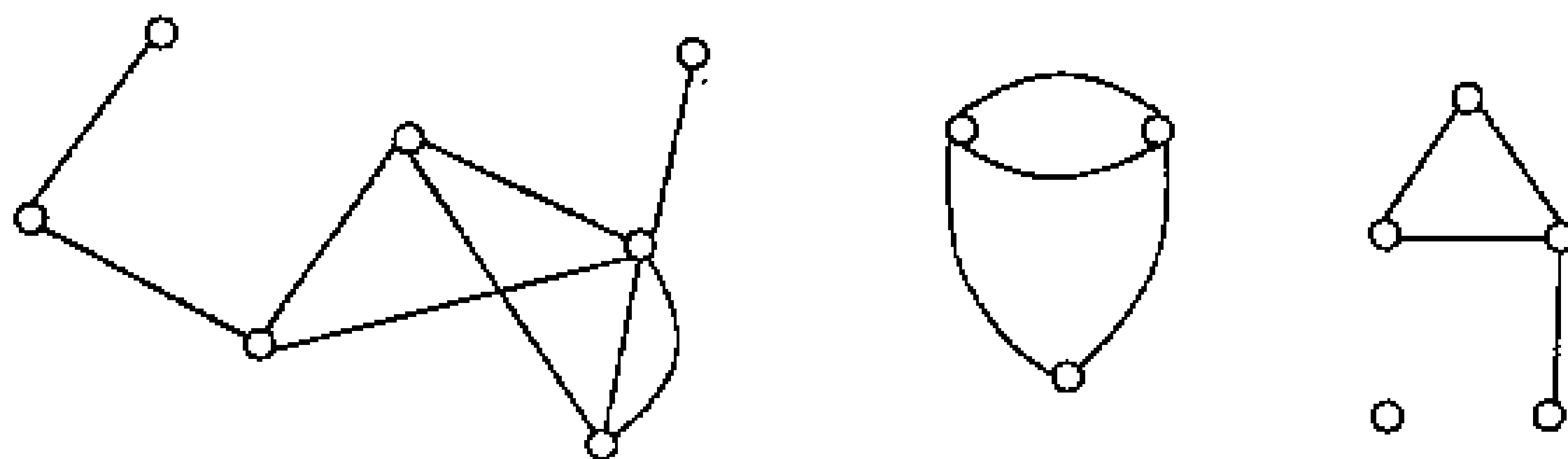


图 1-12

定义 4　如果无向图 G 中每一对不同的顶点 x 和 y 都有一条路(即 $W(G)=1$),则称 G 是**连通图**(**connected digraph or graph**),反之称为**非连通图**(**disconnected graph**).

设 V_1 和 V_2 是 $V(G)$的两个不相交的子集,记

$[V_1,V_2]=\{e\mid e\in E(G),e$ 的两个端点分别在 V_1 和 V_2 中$\}$,显然有下面的结论成立.

引理 1 非平凡图 G 是连通图当且仅当对 $V(G)$的每一个非空真子集 $S,[S,\bar{S}]\neq\emptyset(\bar{S}=V-S)$.

定理 3 设 G 是P 阶连通图,则$|E(G)|\geqslant P-1$.

证明 只需考虑连通的简单图即可.

设 $v_1\in V(G)$,若 G 至少有两个顶点,则令 $V_1=\{v_1\}$,由引理 1 知$[V_1,\bar{V}_1]\neq\emptyset$,令 $e_1\in[V,\bar{V}_1]$,不妨设边 $e_1=\{v_1,v_2\}$. 取 $V_2=\{v_1,v_2\}$,若 V_2 是 $V(G)$的真子集,则存在 $e_2\in[V_2,\bar{V}_2]$,记 v_3 为 e_2 在 $\bar{V}_2$ 中的一个端点.再取 $V_3=\{v_1,v_2,v_3\}$,若 $V_3=V(G)$,结论已成立,否则与上一样可取到一条边 $e_3\in[V_3,\bar{V}_3]$,继续这--过程,便可找出 G 中的$P-1$ 条边.

下面讨论连通图与顶点之间的若干关系.我们把度数为 1 的顶点称为**悬挂点**.

定理 4 设连通图 G 至少有两个顶点,其边数小于顶点数,则此图至少有一个悬挂点.

证明 设 G 是满足定理条件的P 阶图,显然 G 不含孤立顶点.假设 G 没有悬挂点,则对每一个顶点 $v,d(v)\geqslant 2$,于是有 $2|E(G)|=\sum\limits_{v\in V(G)}d(v)\geqslant 2P$,这与$|E(G)|<P$ 矛盾.故 G 中至少有一个悬挂点.

定理 5 设简单图 G 的顶点序列为 $v_1,v_2,\cdots,v_p$,度数依次是 $d(v_1)\leqslant d(v_2)\leqslant\cdots\leqslant d(v_p)$.如果对任意的 $j\leqslant p-\Delta(G)-1$,有 $d(v_j)\geqslant j$,则 G 是连通图.

证明 假设 G 是非连通图,令 G_1 是 G 中不含 v_p 的一个连通分支,$|V(G_1)|=k$,而 G_2 是 G 中含 v_p 的连通分支,则 G_2 至

少有 $d(v_p)+1=\Delta(G)+1$ 个顶点,且

$$|V(G_1)|+|V(G_2)|\leqslant P$$

$$k=|V(G_1)|\leqslant P-V(G_2)\leqslant P-\Delta(G)-1$$

则由已知,$d(v_k)\geqslant k$.若记

$$V(G_1)=\{v_{i_1},v_{i_2},\cdots,v_{i_k}\},i_1<i_2<\cdots<i_k$$

则 $d(v_{i_k})\geqslant d(v_k)\geqslant k$,因而

$$|V(G_1)|\geqslant d(v_{i_k})+1\geqslant k+1$$

这与 $|V(G_1)|=k$ 相矛盾,所以 G 是连通的.证毕

由此定理可得以下推论.

推论 1 设 G 是 P 阶简单图,每个顶点的度至少是 $[\frac{P}{2}]$,则 G 是连通图($[x]$表示不超过 x 的最大整数).

定义 5 设 $D=\langle V(D),E(D),\varphi(D)\rangle$是有向图,$x,y\in V(D)$,若图 D 中存在 x 到 y 的有向路,称结点 x **可达**结点 y.规定 x 到自身总是可达的.

可达性是有向图 D 结点集 $V(D)$上的二元关系,它具有自反性和传递性,未必具有对称性.

定义 6 设 D 是有向图,任何一对结点间,至少有一个结点可达另一个结点,则称这个有向图是**单侧连通的(unilateral connected)**.如果有向图 D 的任何一对结点间是相互可达的,则称这个有向图是**强连通的(strongly connected)**.若有向图 D 的基础图是连通图,则称有向图 D 是**弱连通的(weakly connected)**.

设 $D=\langle V(D),E(D),\varphi(D)\rangle$是有向图,结果点集 $V(D)$上的顶点间的强连通关系显然也是等价关系,它确定 $V(D)$的一个划分$\{V_1,V_2,\cdots,V_m\}$,使得当且仅当两个结点 x 和 y 属于同一个 V_i 时,x 和 y 强连通(即 x 和 y 相互可达).我们称 V_i 在D 中的导出子图$D[V_i]$为 D 的**强连通分支或强分图**.D 的**强连通分**

支数记作 $\vec{W}(D)=m$. 如图 1-13 中的(a)图 D, $\vec{W}(D)=3$, 且强连通分支如图 1-13 中的(b)所示. 若 $\vec{W}(D)=1$, 则称 D 为**强连通图**, 否则称为**非强连通图**.

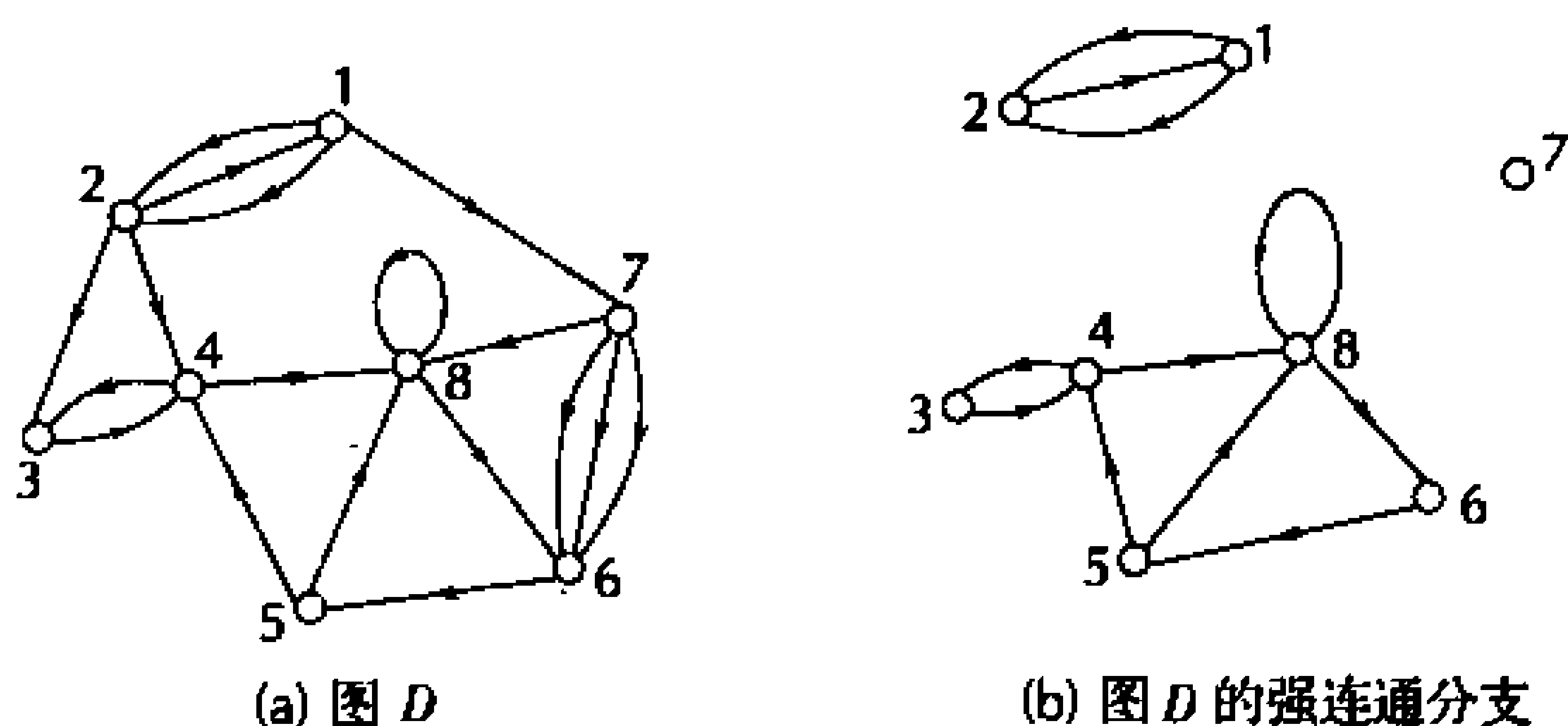


图 1-13

显然, 对于有向图 D, 若 D 是强连通的, 则 D 必是单侧连通的; 若图 G 是单侧 连通的, 则必是弱连通. 这两个命题, 其逆不真.

设 D 是有向图, $x,y\in V(D)$, D 中所有从 x 到 y 的有向路的最小长度称为从 x 到 y 的**距离(distance)**, 记作 $d(x,y)$. 若 x 不可达 y, 则约定 $d(x,y)=\infty$. 距离显然满足下列性质: (1) $d(x,y)\geqslant 0$, (2) $d(x,x)=0$, (3) $d(x,y)+d(y,z)\geqslant d(x,z)$. 有关距离的概念同样适用于无向图. 比如, 设 x,y 为无向图 G 的任两个顶点, G 中所有从 x 到 y 的路的最小长度称为顶点 x 与 y 的距离, 记作 $d(x,y)$ 或 $d(y,x)$. 若 G 中不存在 x 到 y 的路, 则同样有 $d(x,y)=\infty$. 另外, 我们把 $D=\max\{d(x,y)\mid \forall x,y\in V(G)\}$ 称为图 G 的**直径(diameter)**.

第五节　连通度和二分图

定义 1　如果在图 G 中删去一个结点 x 后，图 G 的连通分支数增加，即 $W(G-x)>W(G)$，则称结点 x 为 G 的**割点**(**cut vertex**)．如果在图 G 中删去一条边 e 后，图 G 的连通分支数增加，即 $W(G-e)>W(G)$，则称边 e 为 G 的**割边**(**cut edge**)或**桥**．

定义 2　没有割点的非平凡连通图称为**块**(**block**)．G 中不含割点的极大连通子图称为**图 G 的块**．

例如，图 1-14 所示为图 G 及 G 的五个块 B_1, B_2, B_3, B_4, B_5．

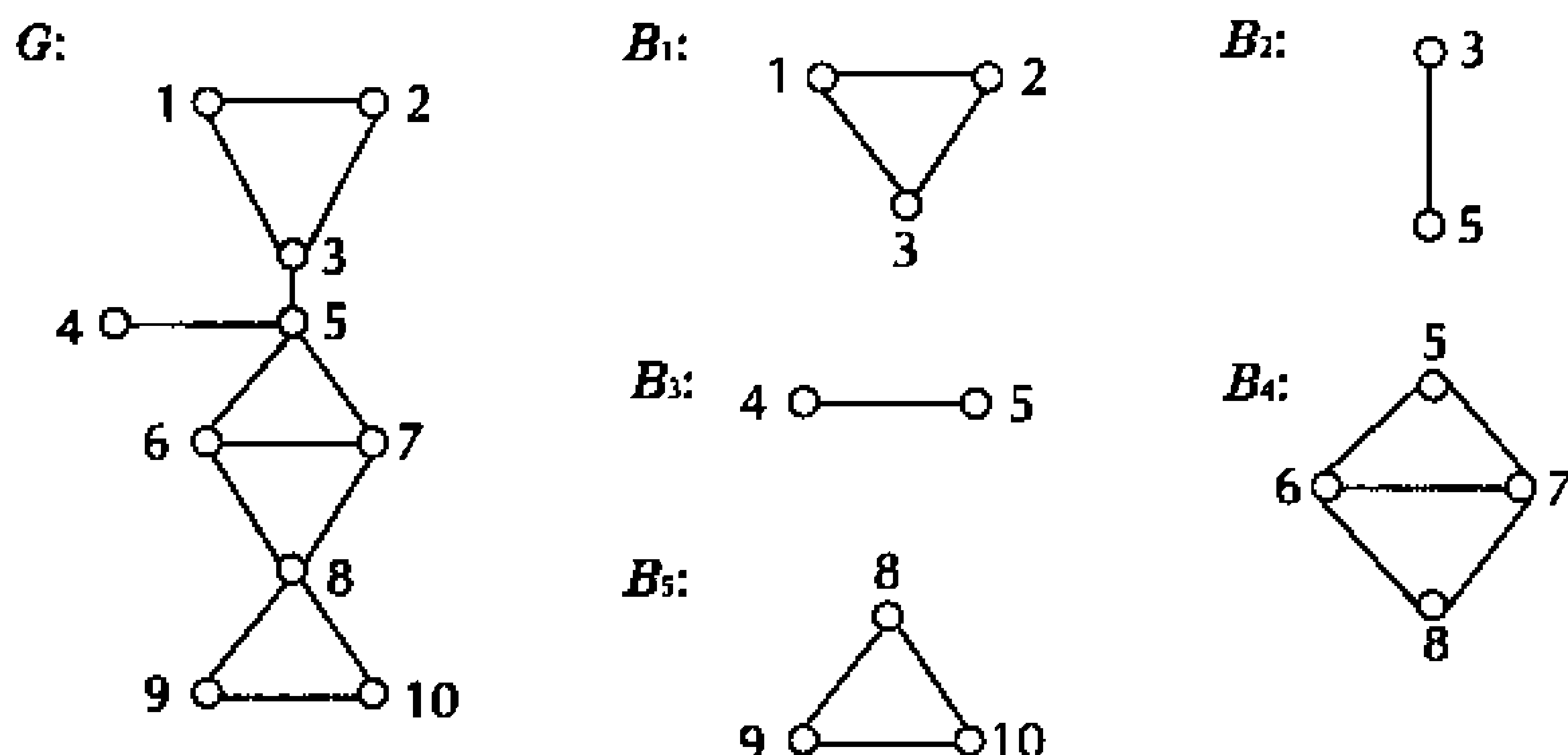


图 1-14

定义 3　如果图 G 的顶点集的一个真子集 T 满足 $G-T$ 不连通或是平凡图，则称 T 为 G 的一个**点割**(**vertex cut**)．如果图 G 的边集的一个真子集 S 满足 $G-S$ 不连通或是平凡图，则称 S 为 G 的一个**边割**(**edge cut**)．

定义 4　设 G 是连通图，称 $K(G)=\min\left\{|T|\,\middle|\,T \text{ 是 } G \text{ 的点割}\right\}$ 为 G 的**点连通度**(**vertex connectivity**)或**连通度**；称 $\lambda(G)=$

$\min\{|S| \mid S$ 是 G 的边割$\}$为 G 的**边连通度**(**edge connectivity**).

定理 1 对一个图 G,有 $K(G) \leqslant \lambda(G) \leqslant \delta(G)$.其中 $\delta(G)$ 是图 G 的最小顶点度.

证明 若 G 不连通,则 $K(G)=\lambda(G)=0$,故上式成立.若 G 连通,

(1)先证 $\lambda(G) \leqslant \delta(G)$

设 x 是 G 中度数最小的顶点,即 $d(x)=\delta(G)$.设所有与 x 关联的边集为 $S(x)$,显然 x 是图 $G-S(x)$的一个孤立点.于是 $\lambda(G) \leqslant |S(x)|=\delta(G)$.

(2)再证 $K(G) \leqslant \lambda(G)$

当 $\lambda(G)=1$ 时,显然 $K(G)=1$.

假设对所有 $\lambda(G)=n(n \geqslant 2)$的图 G,有 $K(G) \leqslant \lambda(G)$.今设 $\lambda(H)=n+1$, S 是 H 的一个边割,且 $|S|=n+1$.若边 $e=\{u,v\} \in S$,易知 $\lambda(G-e)=n$,故由假设知,$K(G-e) \leqslant \lambda(G-e)=n$,并设 T 是 $G-e$ 的一个点割,且 $|T| \leqslant K(G-e)$.而此时 $T \cup \{u\}$就是 H 的一个点割,即 $K(H) \leqslant |T \cup \{u\}| \leqslant n+1=\lambda(H)$.由归纳法原理知,$K(G) \leqslant \lambda(G)$.证毕

如图 1-15 所示图 G 满足:$K(G)=2, \lambda(G)=3, \delta(G)=4$.

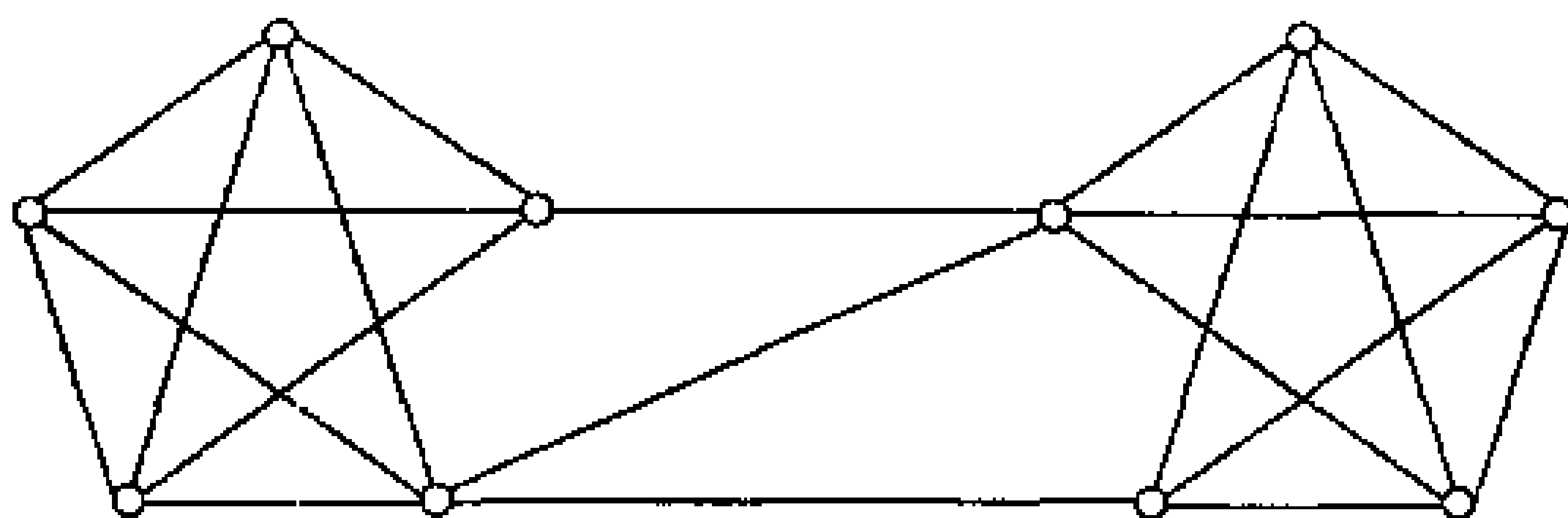


图 1-15

定义 5 如果无向图 G 的连通度 $K(G) \geqslant n(n>1)$,称图 G 是 n **连通的**或 G 为 n **连通图**.若 $\lambda(G) \geqslant n(n>1)$,称图 G 是 n

边连通的或G为n边连通图.

定理 2 设图G是n连通的,$\forall x\in V(G)$,则$\deg(x)\geqslant n$.

证明 假设G有一个顶点y,且$\deg(y)=n-1$,即y与$n-1$条边关联.设与y关联的$n-1$个顶点构成的集合为S,显然S是G的一个点割.因而$K(G)\leqslant|S|=n-1$.这与$K(G)\geqslant n$矛盾.

定理 3 若G是2边连通图,则G有强连通的定向图.

证明 设G是2边连通图,则G必含有圈.先取一个圈C_1,我们归纳地定义G的连通子图序列$G_1,G_2,\cdots$如下:$G_1=C_1$;若$G_i(i=1,2,\cdots)$不是G的生成子图,设v_i是在G中而不在G_i中的一个顶点,则存在从v_i到G_i的边不重路P_i和Q_i,定义

$$G_{i+1}=G_i\cup P_i\cup Q_i$$

由于$|V(G_{i+1})|>|V(G_i)|$,这个序列必然终止于G的一个生成子图G_n.

现依次给每个G_i定向:首先让G_1的定向图$\vec{G}_1$成为一个有向圈;对$G_{i+1}=G_i\cup P_i\cup Q_i$,设已有定向图$\vec{G}_i$,让$P_i$成为以$v_i$为起点的有向路,而$Q_i$成为以$v_i$为终点的有向路,得$\vec{G}_{i+1}$,易见$\vec{G}_{i+1}$是强连通有向图,$i=1,2,\cdots,n$.因此最后的$\vec{G}_n$是强连通有向图.由于$G_n$是$G$的生成子图,所以$G$有强连通的定向图.

显然,一个图G有强连通的定向图的必要条件是G为2边通的.否则G中有割边,这与G有强连通的定向图矛盾.证毕

设G是表示城市街道系统的图,顶点表示交叉路口.如果图G是连通的,则可从城市的任何一点到其他任意一点.我们能否把街道变成单行道路系统,使从城市中的任何点仍有可能沿规定单行方向到达任意其他点,以达到城市交通的畅通,这实际上就是关于图的定向问题.

下面,我们来介绍二分图的概念.

定义 6 若把简单图G的顶点集合分成两个不相交的非空

集合V_1,V_2，使得图G中的每一条边，与其关联的两个结点分别在V_1中和V_2中(因此G里没有边是连接V_1中的两个顶点或V_2里的两个顶点)，则G称为**偶图**或**二分图**(**bipartite graph**)，记作$G=\langle V_1,V_2,E\rangle$，其中$V_1$和$V_2$叫做$G$的二划分．对于二分图$G=\langle V_1,V_2,E\rangle$，若$|V_1|=m$，$|V_2|=n$，且两个顶点之间有一条边，当且仅当一个顶点属于$V_1$而另一个顶点属于$V_2$，则称该图为结点$m$和$n$的**完全偶图**或**完全二分图**(**complete bipartite graph**)，记作$K_{m,n}$．

如图1-16所示，G_1和G_2都是二分图，且$G_1\cong G_2$．图1-17显示完全偶图$K_{2,3}$，$K_{3,3}$，$K_{3,6}$及$K_{2,6}$．

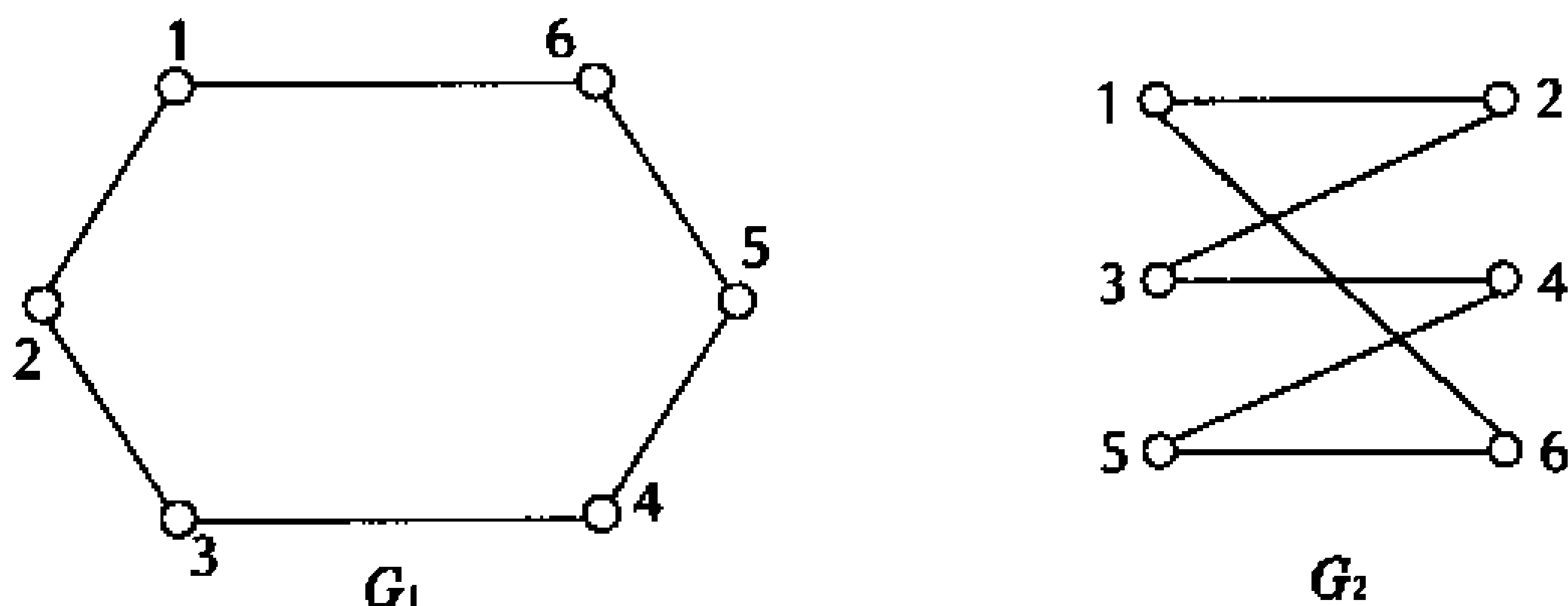


图1-16

例1　试说明n立方体Q_n是二分图．

证明　为方便，记$V(Q_n)=\{x_1x_2\cdots x_n \mid x_i\in\{0,1\},i=1,2,\cdots,n\}$．由$Q_n$的定义知$Q_n$为简单图．而且，若$x=x_1x_2\cdots x_n$，$y=y_1y_2\cdots y_n\in V(Q_n)$，则边$\{x,y\}\in E(Q_n)\Leftrightarrow \sum\limits_{i=1}^{n}|x_i-y_i|=1$(两结点序列恰恰差一位)．

令$X,Y\subset V(Q_n)$，而且$X=\{x_1x_2\cdots x_n \mid x_1+x_2+\cdots+x_n\equiv 0(\mathrm{mod}2)\}$，$Y=\{y_1y_2\cdots y_n \mid y_1+y_2+\cdots+y_n\equiv 1(\mathrm{mod}2)\}$．显然$X\cup Y=V(Q_n)$，$X\cap Y=\varnothing$．而且，若存在$x=x_1x_2\cdots x_n$和$x'=$

$x_1'x_2'\cdots x_n' \in X$,使$\{x, x'\} \in E(Q_n)$,则应有 $\sum_{i=1}^{n} |x_i - x_i'| = 1$,即有$|(x_1 + x_2 + \cdots + x_n) - (x_1' + x_2' + \cdots x_n')| = 1$,这与 $x, x' \in X$ 矛盾.所以 X 中任何两顶点之间无边相连.同理可证 Y 中任何两顶点之间也无边相连.因此 Q_n 是二分图.

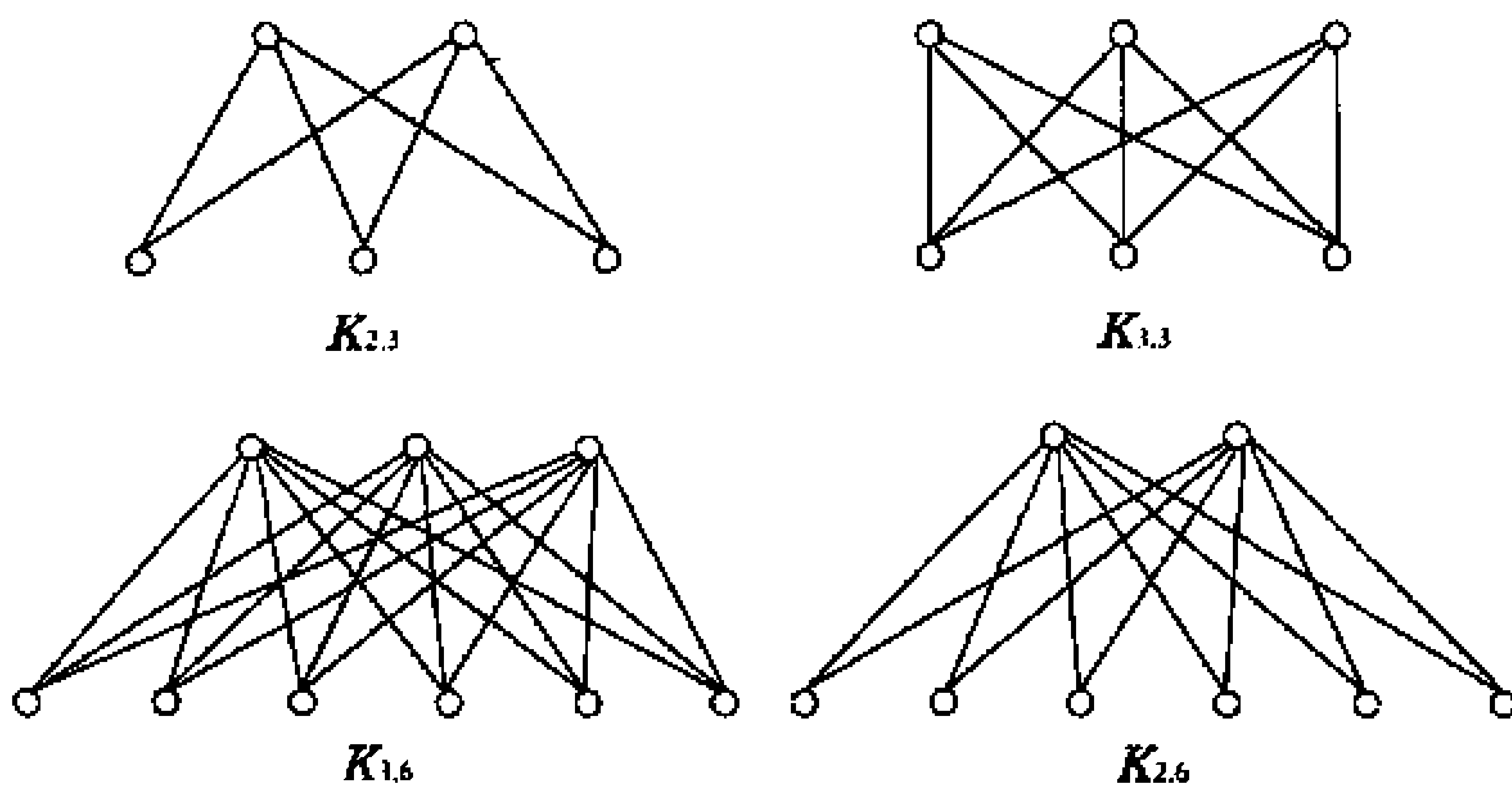


图 1-17

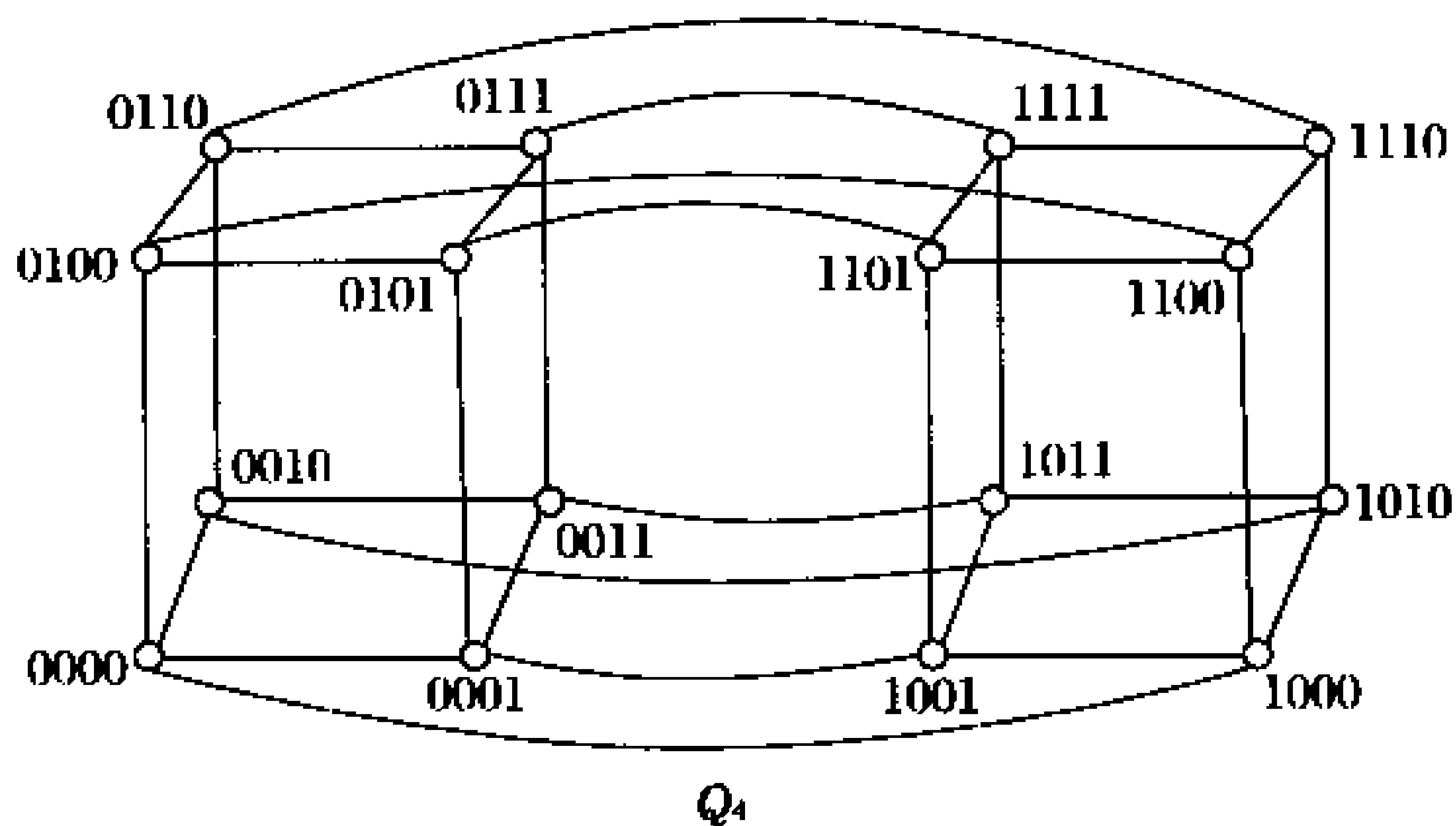
如图 1-18 显示 Q_4 的两个图形表示.

定理 4　非平凡图 G 是二分图当且仅当 G 中不含有长为奇数的圈.

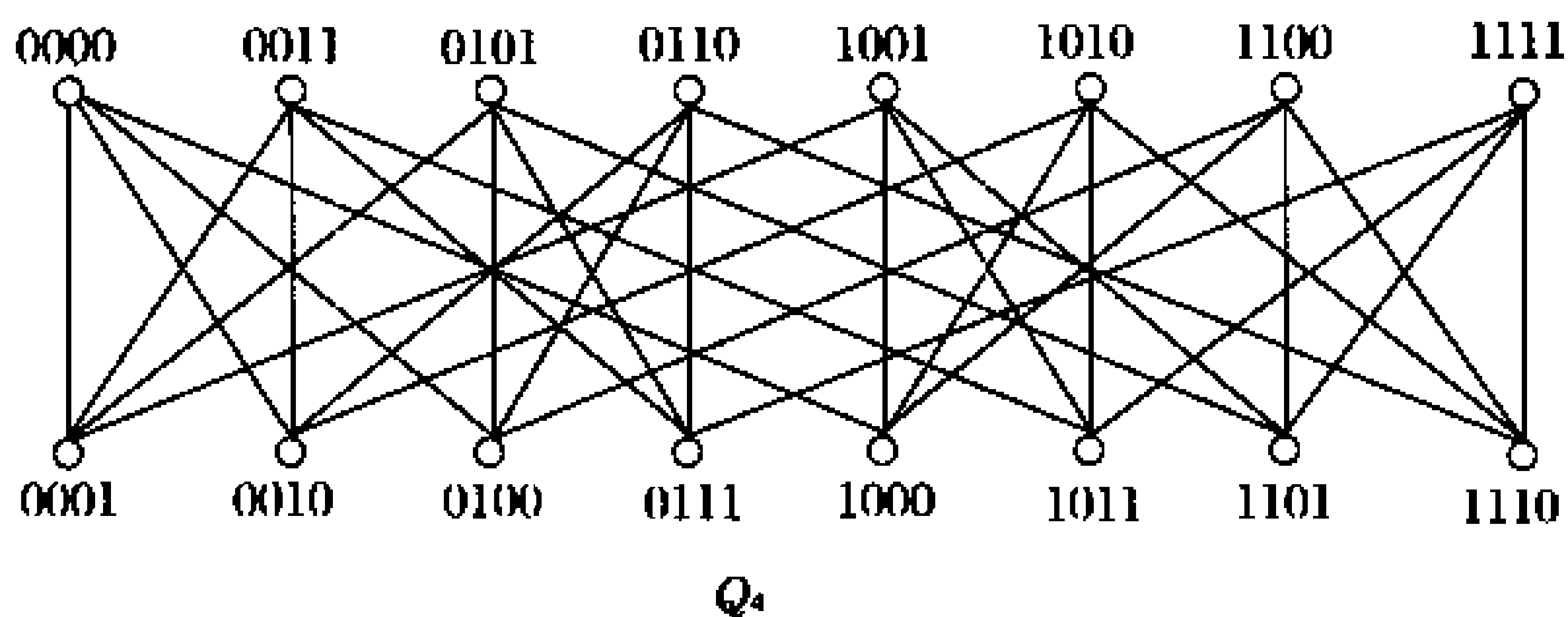
证明(⇒)设 G 是一个二分图,G 的二划分为 V_1 和 V_2,则 $G[V_1]$和 $G(V_2)$为零图.设 $c = v_1 v_2 \cdots v_k v_1$ 是 G 中长度为 k 的一个圈,下证 k 为偶数.

不妨设 $v_1 \in V_1$,由于 v_2 与 v_1 相邻,故 $v_2 \in V_2$;同样因 $v_2 \in V_2$,v_3 与 v_2 相邻,有 $v_3 \in V_1$.一般说来 $v_1, v_3, v_5, \cdots \in V_1$,$v_2, v_4, v_6, \cdots \in V_2$,又因为 $v_1 \in V_1$,边$\{v_1, v_k\} \in E(G)$,所以 $v_k \in V_2$,即得 k 为偶数.

(⇐)不妨设 G 中每一对点之间有路连接(否则只需考虑 G



Q_4



Q_4

图 1-18

的每个每一对点之间有路连接的极大子图). 任取 G 的一个顶点 u,由 G 的假设,对 G 的每个顶点 v,在 G 中存在 $u-v$ 路. 现利用 u 对 G 的顶点进行分类. 设

$$V_1=\{v\mid v\in V(G),G\text{ 中存在一条长为偶数的 }u-v\text{ 路}\}$$

$$V_2=\{v\mid v\in V(G),G\text{ 中存在一条长为奇数的 }u-v\text{ 路}\}$$

显然,$u\in V_1$. 由于图 G 中不存在长度为奇数的圈,所以对任一个点 v,G 中所有从 u 到 v 的路的长度都有相同的奇偶性,因而 $V_1\cap V_2=\varnothing$. 由 G 的假设,$V_1\cup V_2=V(G)$. 现对 G 的每一条边 $e=\{u_1,u_2\}$,若 u_1,u_2 都在 V_1 上,则存在两条路 P_1 与 P_2 分

别连接 u 与 u_1 和 u 与 u_2，且 P_1、P_2 的长度均为偶数，闭链 $P_1 \cup P_2 \cup \{e\}$ 的长度为奇数，则不难看出 G 中有一条长为奇数的圈，矛盾．同样 u_1 和 u_2 不能同时含在 V_2 中．故 e 的两个端点分别在 V_1 和 V_2 中．因此 G 是二分图．证毕

由定理 3 可知图 1-19(a)所示的图不是二分图，因为它包含一个长为 3 的圈 $c = v_4 v_5 v_6 v_4$．图 1-19(b)所示的图是一个二分图，它不含长为奇数的圈．

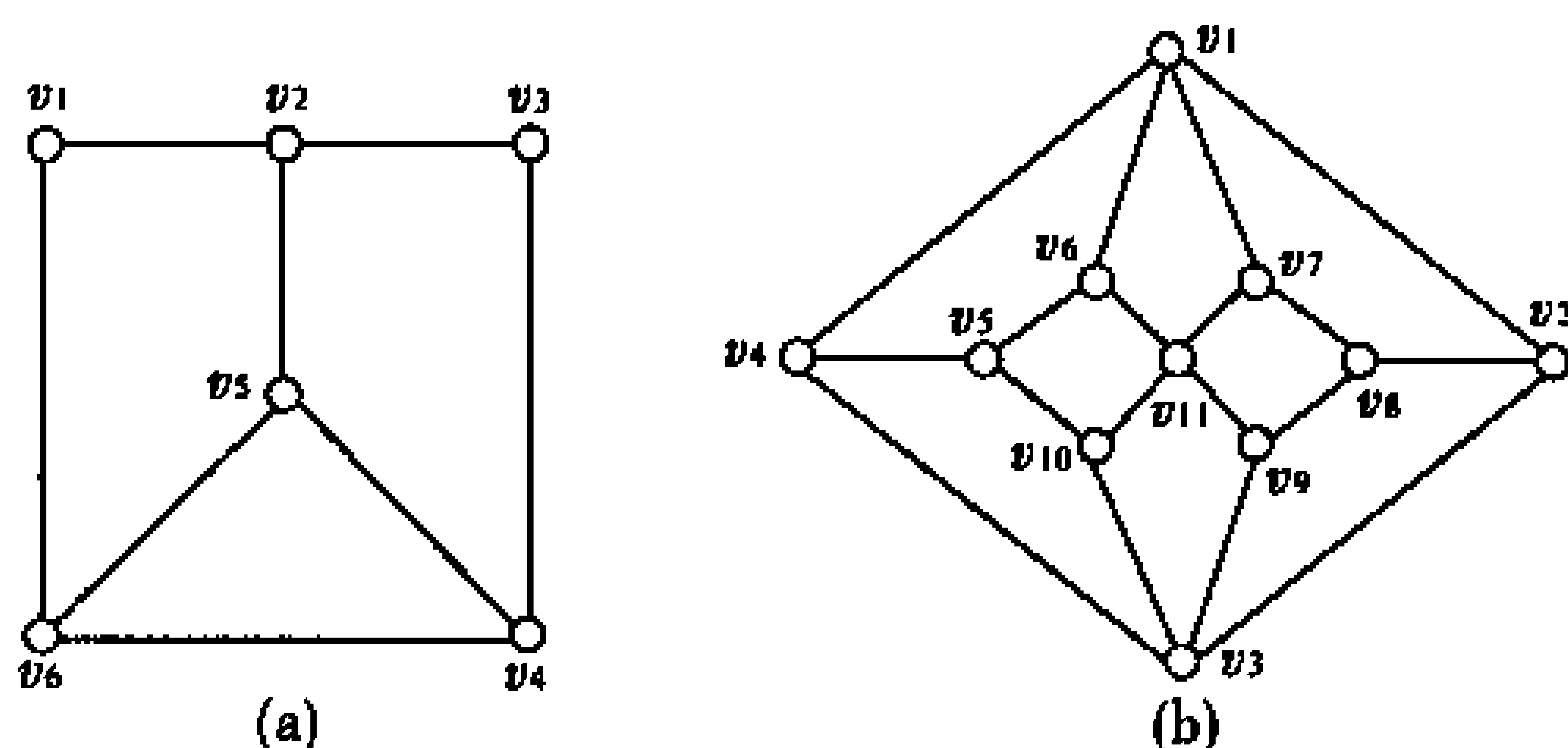


图 1-19

第六节　图的矩阵表示

我们知道图的数学抽象是三元组，还知道形象直观的图的图形表示．为便于计算，特别为便于用计算机处理图，下面介绍图的第三种表示方法——图的矩阵表示．

一、邻接矩阵(adjacency matrix)

设 $G=\langle V,E,\varphi\rangle$是任意图，其中 $V=\{x_1,x_2,\cdots,x_n\}$，$E=\{e_1,e_2,\cdots,e_m\}$，则 n 阶方阵 $A=(a_{ij})$称为 G 的**邻接矩阵**．其中 a_{ij}为图 G 中以 x_i 为起点且以 x_j 为终点的边的数目．

例 1 给出图 1-20 所示的图 G_1、G_2 和 G_3 的邻接矩阵.

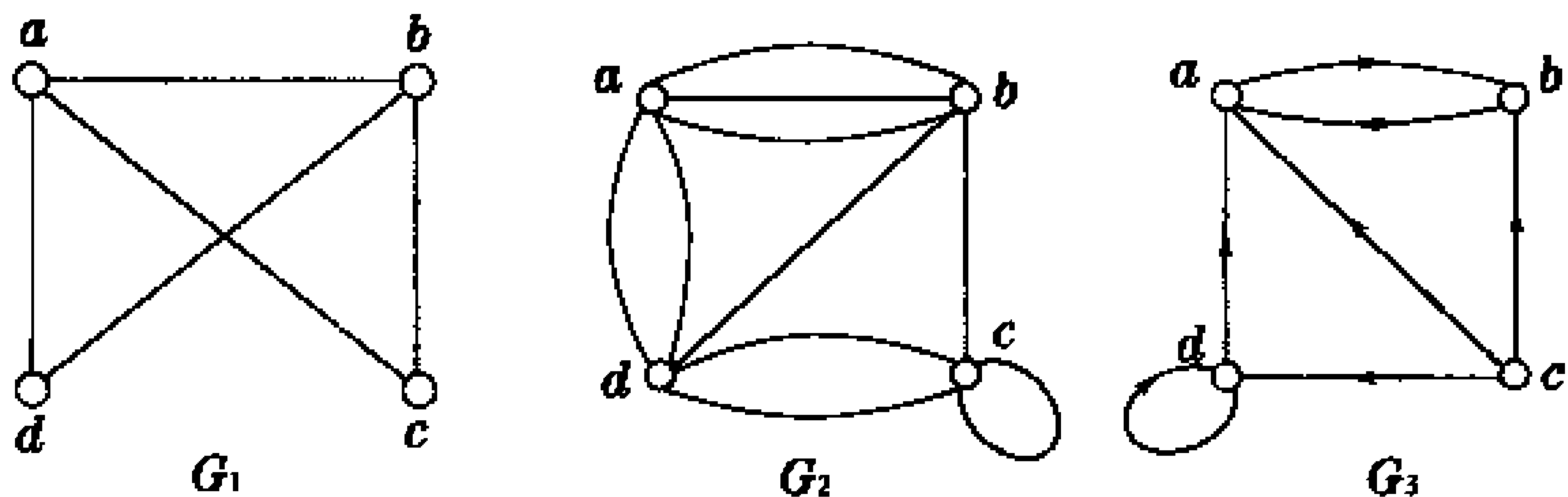


图 1-20

解 图 G_1 的顶点顺序为 a,b,c,d 的邻接矩阵是

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

图 G_2 的顶点顺序为 a,b,c,d 的邻接矩阵是

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

图 G_3 的顶点顺序为 a,b,c,d 的邻接矩阵是

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

由定义知,无向图的邻接矩阵是对称矩阵,而有向图的邻接矩阵未必是对称矩阵. 对于简单无向图 $G=\langle V,E\rangle$,其中 $|V|=n$,则 G 的邻接矩阵是一个 n 阶 0—1 矩阵,若此时图里的边相对少时,邻接矩阵是稀疏矩阵,即只有很少的非 0 项的矩阵,此时,可以

用特殊的方法来表示和计算这样的矩阵.对于简单有向图 $D=\langle V,E\rangle$,结点的邻接关系即是结点集 V 上的二元关系 R,也就是说,图 D 的图形表示即为 R 的关系图,而图 D 的邻接矩阵即为 R 的关系矩阵.

定理 1 已知有向图 $G=\langle V,E,\varphi\rangle$,其中 $V=\{x_1,x_2,\cdots,x_n\}$,且 $A=(a_{ij})_{n\times n}$ 为 G 的邻接矩阵,则 A^k 中的 i 行 j 列元素 $a_{ij}^{(k)}$ 是图 G 中从 x_i 到 x_j 且长度为 k 的有向链的数目.

证明 $k=1$ 时,结论显然成立.

假设 A^{k-1} 的 i 行 j 列元素 $a_{ij}^{(k-1)}$ 为 G 中长度为 $k-1$ 的从 x_i 到 x_j 的有向链的数目.由于 $A^k=A^{k-1}A$,所以我们有 $a_{ij}^{(k)}=\sum_{t=1}^{n}a_{it}^{(k-1)}a_{tj}$.由于 G 中每条从 x_i 到 x_j 且长度为 k 的链都是由长度为 $k-1$ 的从 x_i 到 $x_t(1\leqslant t\leqslant n)$ 的链再接上边 $\langle x_t,x_j\rangle$ 而得到.故由归纳假设知定理得证.

注释 1 该定理同样适合于无向图,而且定理中的链不能改成迹或者路.

例 2 已知图 G 如图 1-21 所示.

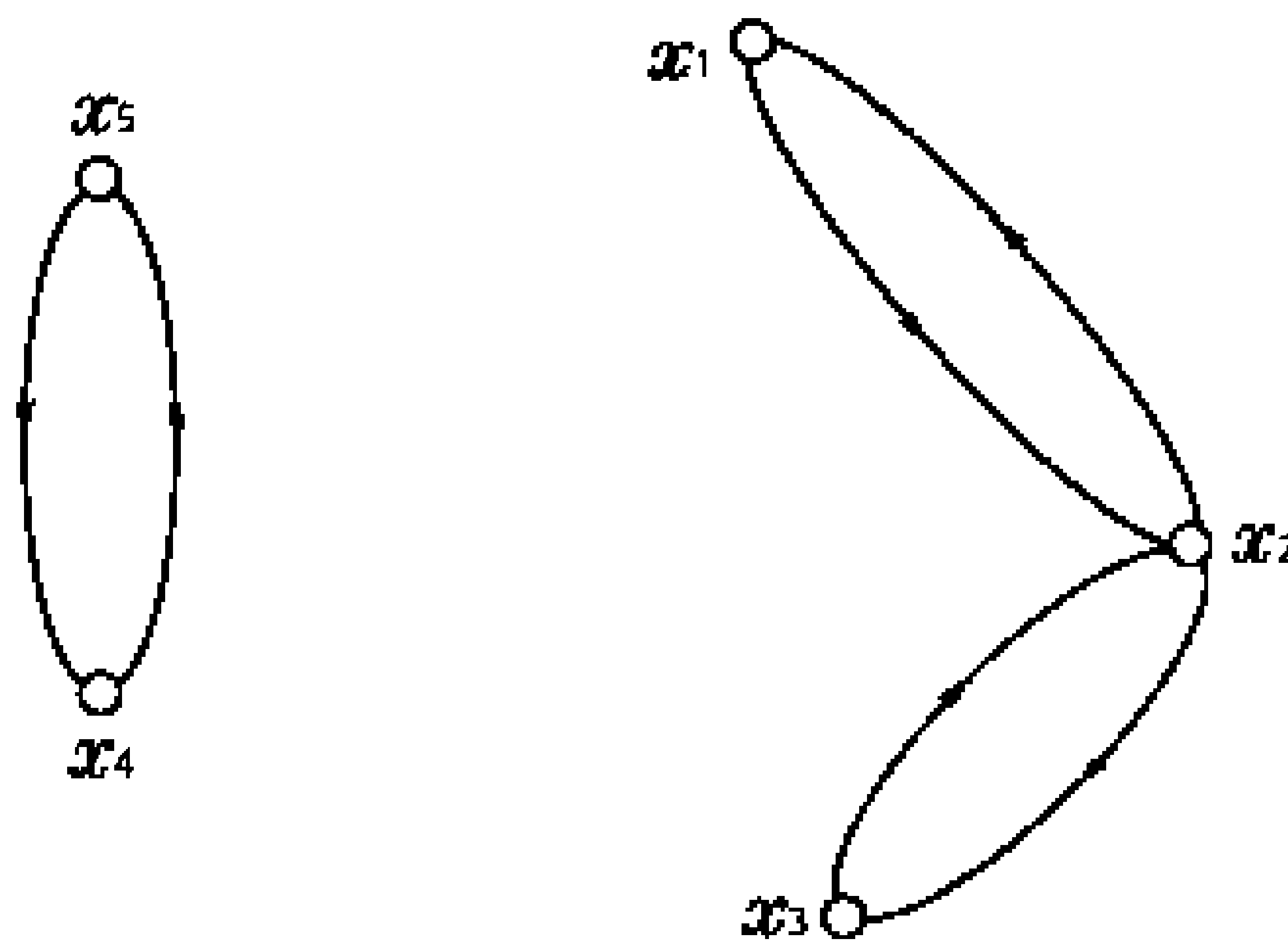


图 1-21

G 的顶点顺序为 x_1,x_2,x_3,x_4,x_5 的邻接矩阵为

$$A=\begin{bmatrix}0&1&0&0&0\\1&0&1&0&0\\0&1&0&0&0\\0&0&0&0&1\\0&0&0&1&0\end{bmatrix}$$

通过矩阵的乘法运算得

$$A^4=\begin{bmatrix}2&0&2&0&0\\0&4&0&0&0\\2&0&2&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$$

由上述矩阵可知，从 x_1 到 x_2 有四条长度为 4 的有向链.

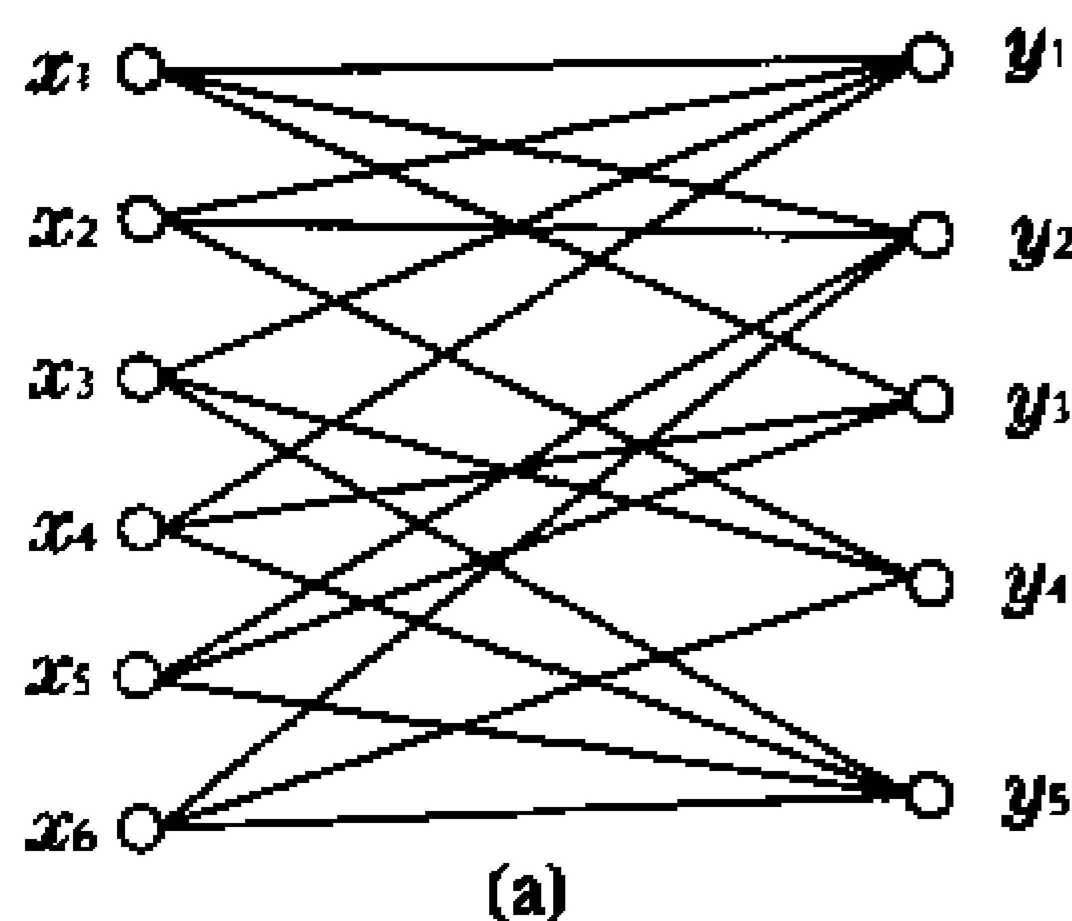
对于一个二分图 $G=\langle V_1,V_2,E\rangle$ 来说，我们可以用一个阶数比邻接矩阵更小的矩阵来表示. 设 $V_1=\{x_1,x_2,\cdots,x_n\}$，$V_2=\{y_1,y_2,\cdots,y_m\}$，作 $n\times m$ 阶矩阵 $A=(a_{ij})_{n\times m}$，其中 a_{ij} 表示 x_i 与 y_j 之间连接的边数.

例 3 图 1-22(a)所示的二分图所对应的 6×5 阶矩阵 A 为

$$A=\begin{bmatrix}1&1&1&0&0\\1&1&0&1&0\\1&0&0&1&1\\1&0&1&0&1\\0&1&1&0&1\\0&1&0&1&1\end{bmatrix}$$

反之，给定一个 0-1 矩阵，能唯一确定一个简单二分图 $G=\langle V_1,V_2,E\rangle$. 例如 0-1 矩阵 B 对应一个二分图 G（如图 1-22(b)所示）为

$$B=\begin{bmatrix}0&1&1&1\\1&0&1&0\\0&0&1&0\\1&1&0&1\end{bmatrix}$$



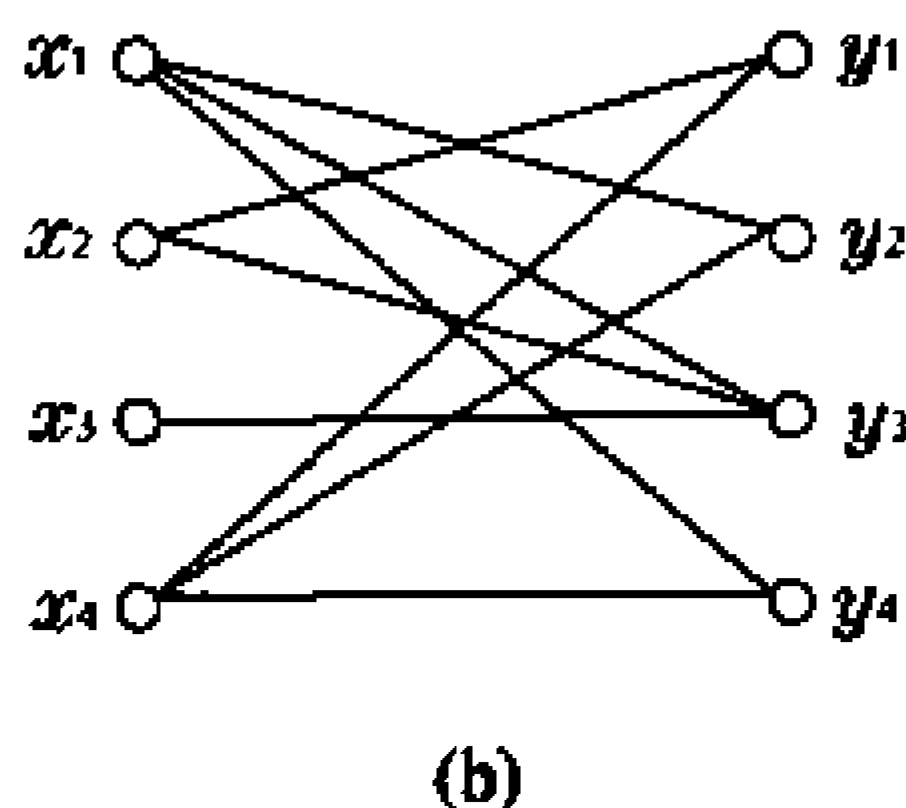


图 1-22

推论 1 若 G 是 P 阶简单图，且 G 的邻接矩阵为 $A=(a_{ij})$，则对 G 的每一个顶点 v_i，$i=1,2,\cdots,p$，有

$$d(v_i)=a_{ii}^{(2)}\text{，其中 } A^2=(a_{ij}^{(2)})$$

证明 图 G 中与顶点 v_i 关联的边数等于从 v_i 到 v_i 长为 2 的链数目，故由定理 1 知结论成立.

给定 P 阶图 G 的邻接矩阵 A，作一个 P 阶方阵：

$$R=A+A^2+\cdots+A^{P-1}=(r_{ij})_{p\times p}$$

显然，r_{ij} 就是 G 中连接顶点 x_i 和 x_j 长度不超过 $P-1$ 的链（若 G 是有向图，则是从 x_i 到 x_j 的有向链）的数目，因而可以利用 R 来判断一个图的连通性.

定理 2 已知 $P(P\geqslant 3)$ 阶图 G 的邻接矩阵为 A，作 P 阶方阵 $R=A+A^2+\cdots+A^{P-1}$，则图 G 连通的充分必要条件为 R 中的每个元素都不等于零.

证明 设 $A=(a_{ij})$，$A^k=(a_{ij}^{(k)})$，$R=(r_{ij})$

(⇒)设 G 是连通图，则 G 中任意两个不同的顶点 x_i 与 x_j 之间有一条路连接. 若记这条路的长度为 k，显然 $k \leqslant P-1$. 则 $r_{ij} \geqslant a_{ij}^{(k)} \geqslant 1$. 而对于任意的 $i(1 \leqslant i \leqslant P)$，因 G 连通，且 $P \geqslant 3$，由推论 1 得 $r_{ii} \geqslant a_{ii}^{(2)} = d(x_i) > 0$，所以 R 没有零元素.

(⇐)设 x_i 与 x_j 是 G 中任意两个不同的顶点. 因为 $r_{ij} = a_{ij} + a_{ij}^{(2)} + \cdots + a_{ij}^{(P-1)} \neq 0$，存在 $1 \leqslant k \leqslant P-1$，使得 $a_{ij}^{(k)} \neq 0$，因而在 G 中有一条长为 k 的链连接 x_i 和 x_j，从而 x_i 与 x_j 之间有一条路. 即图 G 是连通图.

二、关联矩阵(incidence matrix)

设 $D = \langle V, E, \varphi \rangle$ 是有向图，且 $V = \{x_1, x_2, \cdots, x_n\}$，$E = \{e_1, e_2, \cdots, e_m\}$，称 $n \times m$ 阶矩阵 $M = (m_{ij})$ 为有向图 D 的**关联矩阵**，其中

$$m_{ij} = \begin{cases} -2 & e_j \text{ 是自环，且关联于 } x_i \\ 1 & \text{在 } D \text{ 中 } e_j \text{ 以 } x_i \text{ 为起点，} e_j \text{ 不是自环} \\ -1 & \text{在 } D \text{ 中 } e_j \text{ 以 } x_i \text{ 为终点，} e_j \text{ 不是自环} \\ 0 & e_j \text{ 与 } x_i \text{ 不关联} \end{cases}$$

设 $G = \langle V, E, \varphi \rangle$ 为无向图，且 $V = \{x_1, x_2, \cdots, x_n\}$，$E = \{e_1, e_2, \cdots, e_m\}$，称 $n \times m$ 阶矩阵 $M = (m_{ij})$ 为无向图 G 的关联矩阵，其中

$$m_{ij} = \begin{cases} 2 & e_j \text{ 关联于 } x_i, e_j \text{ 是自环} \\ 1 & e_j \text{ 关联 } x_i, e_j \text{ 不是自环} \\ 0 & e_j \text{ 不关联 } x_i \end{cases}$$

例 4 给出图 1-23 所示图 G 和图 D 的关联矩阵.

解 图 G 的关联矩阵为

$$
\begin{array}{c}
\\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\
\begin{bmatrix}
2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

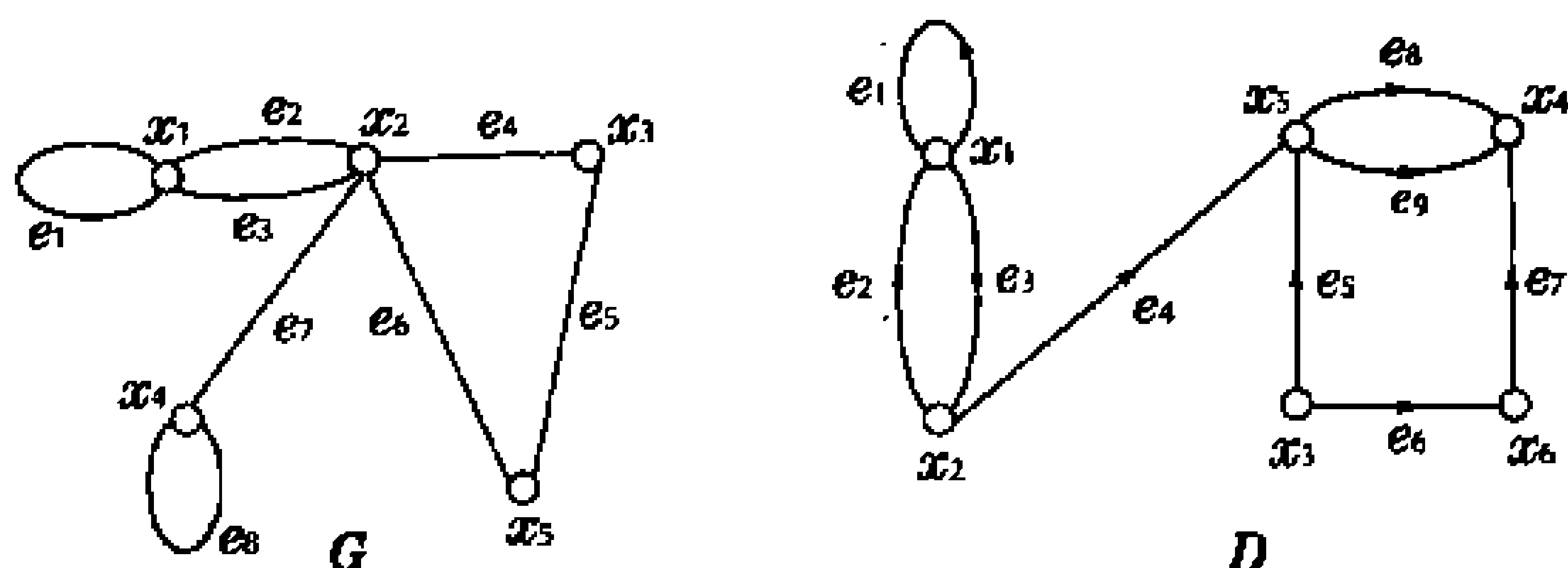


图 1-23

图 D 的关联矩阵为

$$
\begin{array}{c}
\\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{array} \\
\begin{bmatrix}
-2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\
0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

从关联矩阵可以看出无向图的一些性质：

(1)因为每条边关联两个顶点,所以关联矩阵的每一列只有两个1.

(2)关联矩阵的每一行中元素之和为对应顶点的度.

(3)若某行中元素全为0,则对应的顶点为孤立点.

(4)重边所对应的列完全相同.

关于有向图的一些性质可类似地推出.

三、可达矩阵(accessible matrix)

设 $G=\langle V,E\rangle$是无重边有向图,其中 $V=\{x_1,x_2,\cdots,x_n\}$,称 $n\times n$ 阶矩阵$P=(P_{ij})$为 G 的**可达矩阵**,其中

$$P_{ij}=\begin{cases}1 & \text{从 } x_i \text{ 到 } x_j \text{ 至少有一条有向链}\\ 0 & \text{从 } x_i \text{ 到 } x_j \text{ 没有有向链}\end{cases}$$

例 5 无重边有向图 $G=\langle V,E\rangle$,如图 1-24 所示,其顶点顺序为 v_1,v_2,v_3,v_4,v_5 的可达矩阵 P 为:

$$P=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&1\end{bmatrix}$$

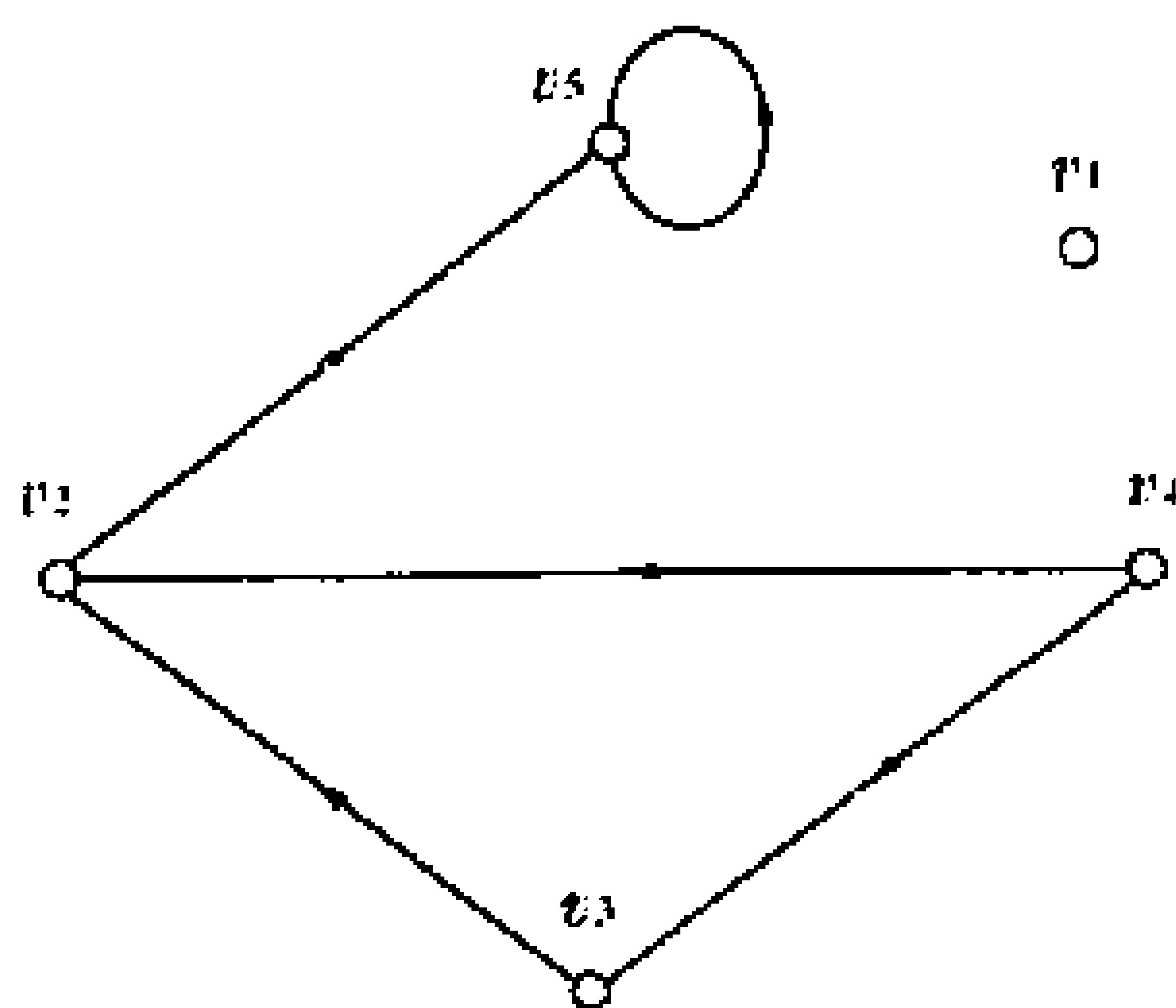


图 1-24

一般地讲,可由图 G 的邻接矩阵A 得到可达矩阵P,即令 $R=A+A^2+\cdots+A^n$,再从 R 中将不为零的元素均改换为 1,而为零的元素不变,这个改换所得的矩阵即为可达矩阵 P.

例 6 对图 1-24 所示的无重边有向图 $G=\langle V,E\rangle$,其邻接矩

阵(顶点顺序为 v_1,v_2,v_3,v_4,v_5)A 为

$$A=\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&0&1\end{bmatrix}$$

$$A^2=\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&0&1&1\end{bmatrix}\qquad A^3=\begin{bmatrix}0&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&1&0&1\end{bmatrix}$$

$$A^4=\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&1&1\end{bmatrix}\qquad A^5=\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&1&1\end{bmatrix}$$

故 $R=A+A^2+A^3+A^4+A^5=\begin{bmatrix}0&0&0&0&0\\0&1&2&2&0\\0&2&1&2&0\\0&2&2&1&0\\0&2&2&3&5\end{bmatrix}$

可得 G 的可达矩阵 $P=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&1\end{bmatrix}$

上述计算可达矩阵的方法还是比较复杂,因为可达矩阵是 0 -1 矩阵(布尔矩阵),由于在可达矩阵 P 中,只关心结点 x_i 到 x_j

是否有有向链，至于有向链的长度与有几条有向链是不被关注的，因此，我们将矩阵 $A, A^2, \cdots, A^n$ 分别改为布尔矩阵 $A^{(1)}, A^{(2)}, \cdots, A^{(n)}$，故可达矩阵 $P=A^{(1)} \vee A^{(2)} \vee \cdots \vee A^{(n)}$，其中 $A^{(k)}$ 表示在布尔运算意义下 A 的 k 次方.

例 7 已知图 1-24 所示图 G 的邻接矩阵 A，

$$A^{(2)}=\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&0&1\end{bmatrix}=\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&0&1&1\end{bmatrix}$$

$$A^{(3)}=\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&0&1&1\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&0&1\end{bmatrix}=\begin{bmatrix}0&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&1&1&1\end{bmatrix}$$

同理可得

$$A^{(4)}=\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&1&0&1&1\end{bmatrix}\qquad A^{(5)}=\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&1&1\end{bmatrix}$$

因而 $P=A^{(1)} \vee A^{(2)} \vee A^{(3)} \vee A^{(4)} \vee A^{(5)}=\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&1\end{bmatrix}$

无重边有向图 $G=\langle V,E\rangle$ 的邻接矩阵是结点集 V 上的二元关系 R 的关系矩阵，则可达矩阵 P 即为 R 的传递闭包 $t(R)$ 的关

系矩阵，因此，可用 Warshall 算法求可达矩阵 P.

习 题 一

1. 在平面上有 n 个点 $S=\{x_1,x_2,\cdots,x_n\}$，其中任两个点之间的距离至少是 1. 证明在这 n 个点中，距离为 1 的点对数不超过 $3n$.

2. 在某次会议中有 n 人参加，其中有些人互相认识，但每两个互相认识的人，都没有共同的熟人，每两个互不认识的人都恰好有两个共同的熟人. 证明每一个参加者都有同样数目的熟人.

3. 证明每个碳氢化合物的分子所含的氢原子数是偶数.

4. 在一次围棋擂台赛中，双方各出 n 名选手. 比赛的规则是双方先各自排定一个次序，设甲方排定的次序为 $x_1,x_2,\cdots,x_n$，乙方排定的次序为 $y_1,y_2,\cdots,y_n$. x_1 与 y_1 先比赛，胜的一位与对方输的下一位选手比赛. 按这种方法进行比赛，直到有一方的最后一位选手出场比赛并且输给对方，比赛就结束. 问最多进行几场比赛可定其胜负（假定比赛不出现平局）.

5. 在一个化学实验室里，有 n 个药箱，其中每两个不同的药箱恰有一种相同的化学品，而且每种化学品恰好在两个药箱中出现，问：

(1)每个药箱有几种化学品？

(2)这 n 个药箱中共有几种不同的化学品？

6. 在一次舞会中，A、B 两国留学生各 $n(n>2)$ 人，A 国每个学生都与 B 国一些（不是所有）学生跳过舞. B 国每个学生至少与 A 国一个学生跳过舞. 证明一定可以找到 A 国两个学生 a_1、a_2 及 B 国两个学生 b_1、b_2，使得 a_1 和 b_1，a_2 和 b_2 跳过舞，而 a_1 和 b_2，a_2 和 b_1 没有跳过舞.

7.(过河问题)一个摆渡人,要把一条狼、一只羊和一捆草从河西运到河东去,由于船太小,除摆渡人之外,一次只能运一个“乘客”.显然,摆渡人不能让狼和羊单独留在岸边,也不能让羊和草单独留在岸边.问摆渡人怎样才能把它们运到河东去?

8. 在 $n\times n$ 的棋盘上,填上 $1\sim n^2$ 的所有自然数.证明总能找到相邻的两个方格(具有公共边的两个方格称为是相邻的),里面所填的两数之差不小于 $\left\{\frac{n+1}{2}\right\}$($\{x\}$表示不小于 x 的最小整数).

9. 任意给定 $(n+1)^2$ 项的递增的自然数列.则下面的结论中必有一条是成立的.

(1)存在 $n+3$ 项的子列,使任一项能整除此子列中它后面的每一项;

(2)存在 $n+1$ 项的子列,使此子列中任一项不能整除它后面的任何一项.

10. 用一些圆面覆盖平面上取定的 $2n$ 个点.试证:若每个圆面至少覆盖 $n+1$ 个点,则任两个点能由平面上的一条折线所连接,而这条折线整个地被某些圆面所覆盖.

11. 若简单图 G 同构于 G 的补图 $\overline{G}$,则称 G 为自补图.

(1)证明自补图的阶数 $n=4k$ 或 $n=4k+1$,k 为某个自然数.

(2)找出所有 4 阶和 5 阶的自补图.

12. 图 G 如图 1-25 所示:

(1)画出下列几个子图:

(a)$G-\{v_1,v_2,v_4\}$

(b)$G[v_1,v_2,v_3]$

(c)$G[e_3]$

(d)$G[e_1,e_2,e_3]$

(e)$G-\{e_4,e_7,e_8,e_9\}$

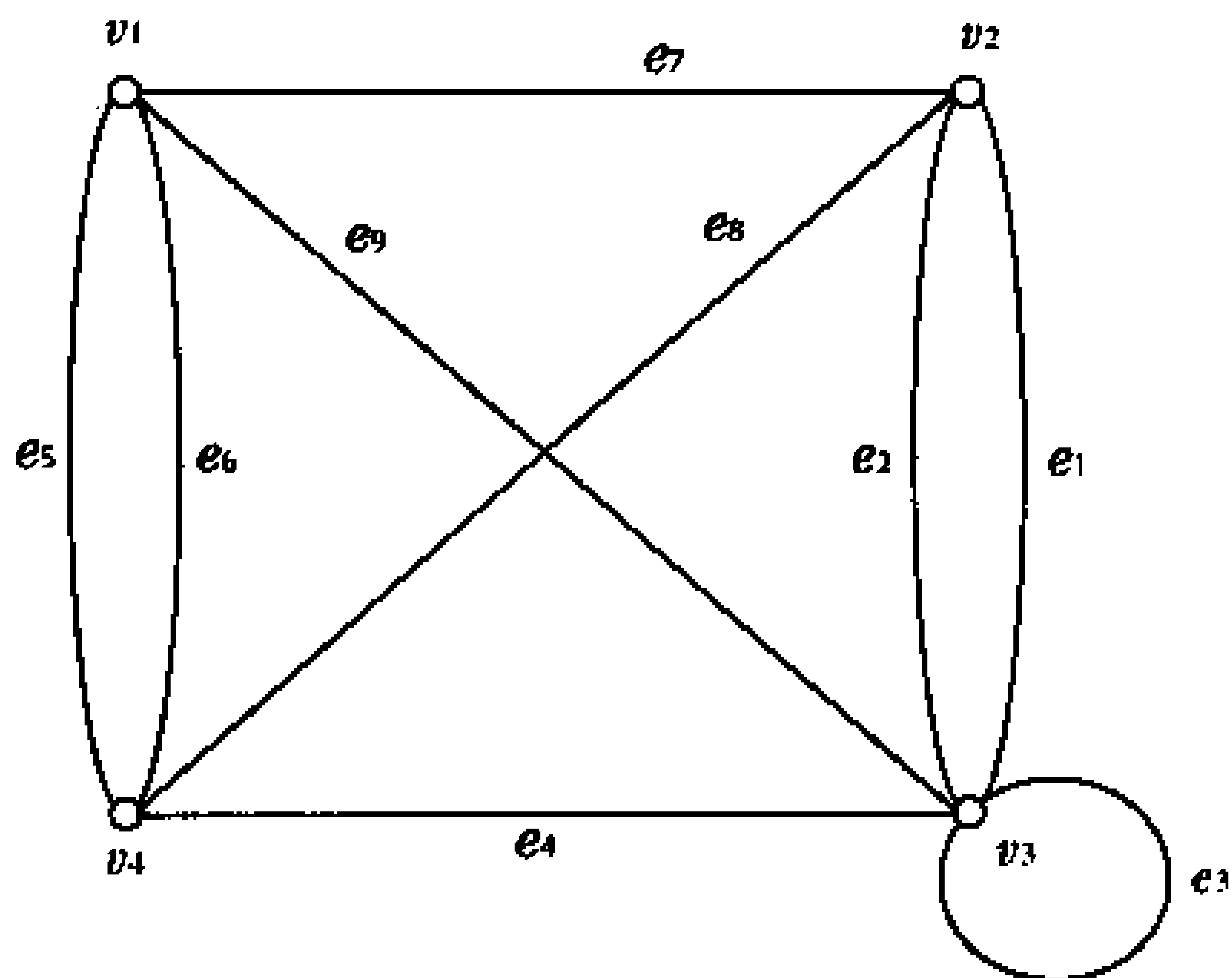


图 1-25 G

(2)上面的 5 个子图中,哪几个子图是正则图?

(3)求 $G[v_1,v_2,v_3]\cup G[v_2,v_3,v_4]$.

13. 已知 n 阶无向简单图 G 有 m 条边,各结点的度数均为 3,(1)若 $m=3n-6$,证明 G 在同构意义下唯一,并求 m,n.(2)若 $n=6$,证明 G 在同构意义下不唯一.

14. 无向图 G 有 21 条边,12 个 3 度数结点,其余结点的度数均为 2,求 G 的阶数 n.

15. 证明若无向图 G 是不连通的,则 G 的补图 $\overline{G}$ 是连通的.

16. 设 G 为 n 阶简单无向图,$n>2$ 且 n 为奇数,G 和 G 的补图 $\overline{G}$ 中度数为奇数的顶点个数是否一定相等? 试证明你的结论.

17. 设简单有向图 G 如图 1-26 所示,试求图 G 的一切强分图,单向分图,弱分图.

18. 下列非负整数序列哪些是度序列? 哪些是图序列?

(1)(1,1,1,2,3)

(2)(0,1,1,2,3,3)

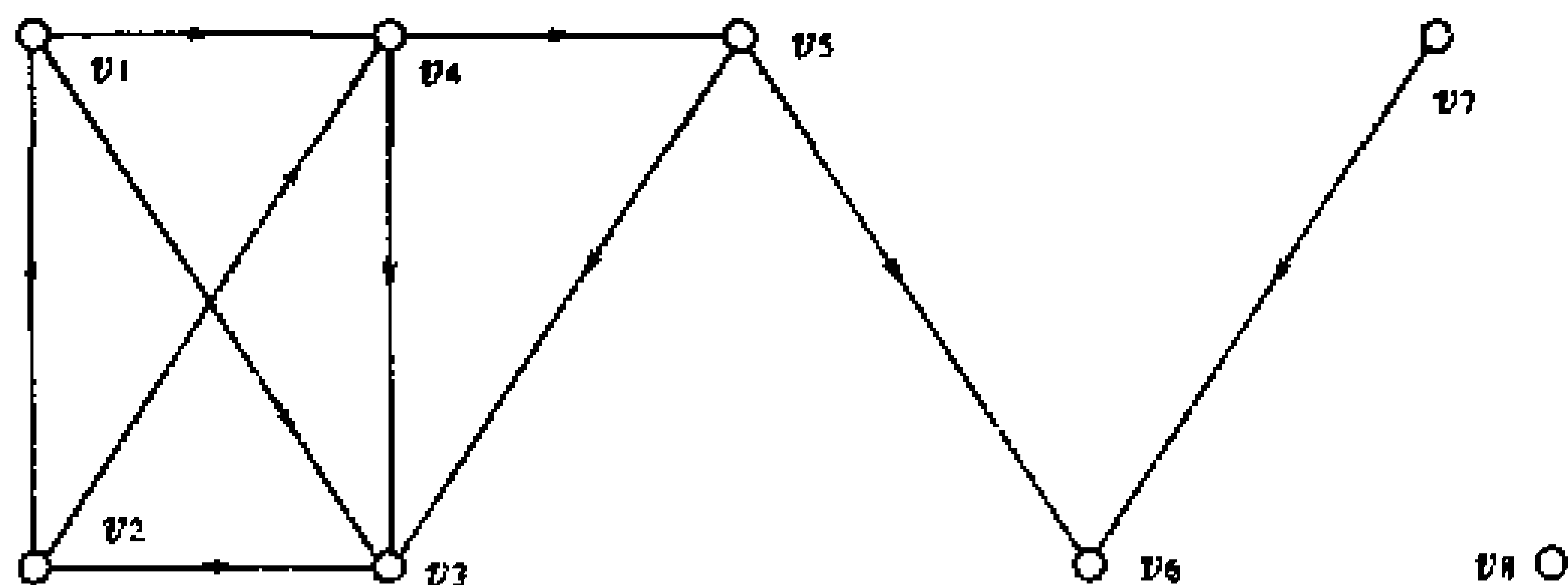


图 1-26　G

(3)(3,3,3,3)

(4)(2,3,3,4,4,5)

(5)(2,3,4,4,5)

(6)(2,2,2,2,2)

(7)(2,3,3,4,5,6)

(8)(1,3,3,4,5,6,6)

19. 画出 4 阶完全图 K_4 的所有非同构的生成子图.

20.9 个数学家在一次国际数学会议上相遇,发现他们中的任意 3 个人中,至少有两个人可以用同一种语言对话.如果每个数学家最多可说三种语言,证明至少有三个数学家可以用同一种语言对话.

21. 已知有向图 G 如图 1-27 所示.

(1)求出 G 的邻接矩阵 A.

(2)求出 A^2,A^3,A^4,说明从 v_1 到 v_4 长度为 1,2,3,4 的有向链各有几条?

(3)求出 A^T,A^TA,AA^T,说明 AA^T 和 A^TA 中第(2,3)个元素和第(2,2)个元素的意义.其中 A^T 为矩阵 A 的转置矩阵.

(4)求出 $A^{(2)},A^{(3)},A^{(4)}$ 及可达矩阵 P.

(5)求出强分图.

22. 已知 A 为 n 阶 0－1 矩阵(布尔矩阵),证明对任何正整数

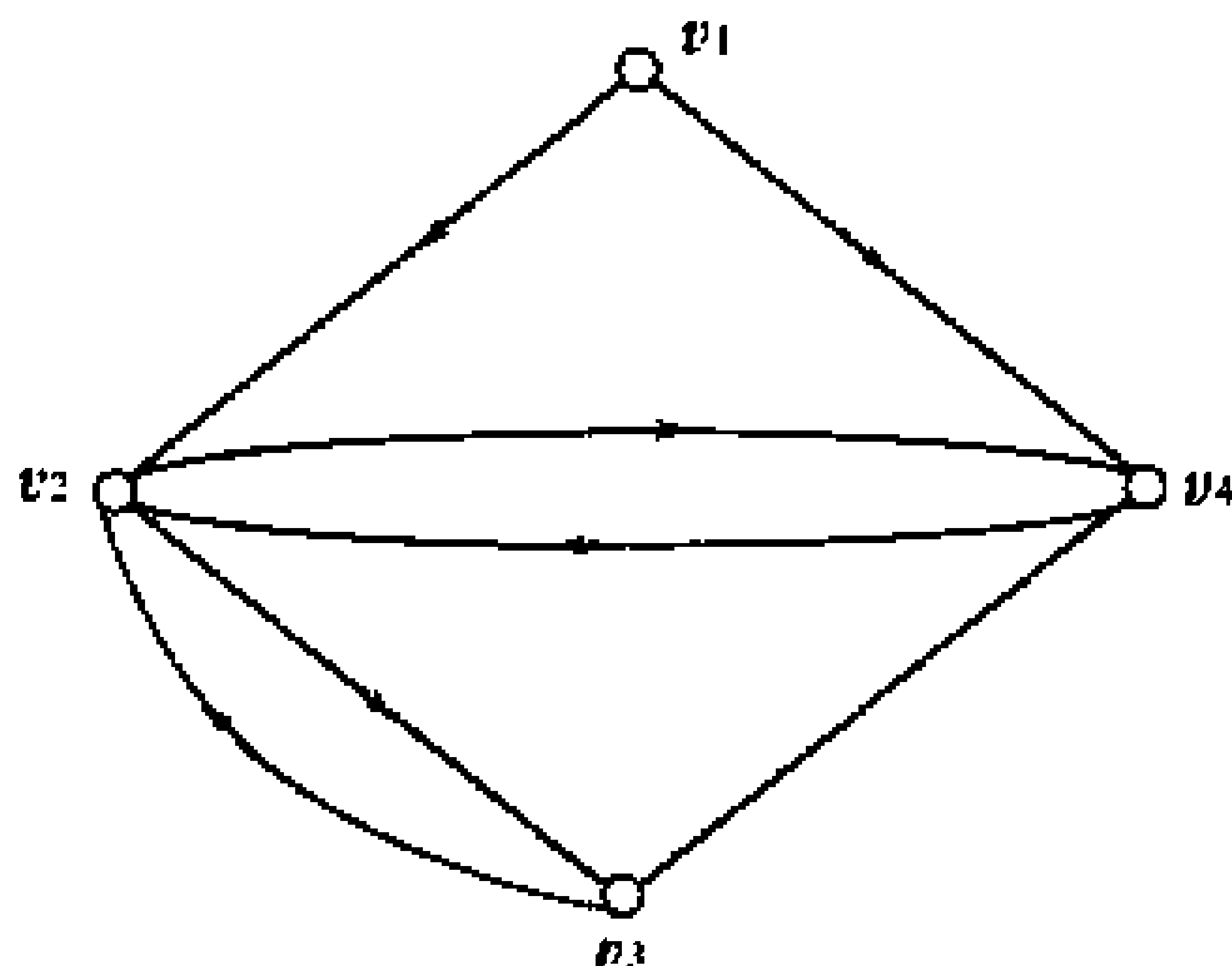


图 1-27　G

k,有$(I \vee A)^k = I \vee A \vee A^2 \vee \cdots \vee A^k$,其中 I 为单位矩阵.

23. 设已知下列事实:a 会讲英语,b 会讲汉语和英语,c 会讲英语、意大利语和俄语,d 会讲日语和汉语,e 会讲德语和意大利语,f 会讲法语、日语和俄语,g 会讲法语和德语.试问这 7 个人中,是否任意两人都能交谈(必要时可借助于其余 5 人组成的译员链).

24. **完全 m 分图** $K_{n_1,n_2,\cdots,n_m}$ 的顶点集合划分成 m 个子集合,各有 $n_1,n_2,\cdots,n_m$ 个元素,而且顶点相邻当且仅当它们属于这个划分的不同子集合.

(1)完全 m 分图 $K_{n_1,n_2,\cdots,n_m}$ 有多少个顶点和多少条边?

(2)给出图 $K_{1,2,3}$,$K_{2,2,2}$,$K_{1,2,2,3}$ 的图形表示.

25. 编写具有下列输入与输出的程序.

(1)给定无向图的各边所关联的顶点对,确定每个顶点的度.

(2)给定有向图的各边所关联的有序顶点对,确定每个顶点的入度和出度.

(3)给定图的各边所关联的顶点对,构造这个图的邻接矩阵.

(4)给定图的邻接矩阵,列出这个图的各边,并且给出每条边出现的次数.

(5)给定无向图各边所关联的顶点对，以及每条边出现的次数，构造这个图的关联矩阵.

(6)给定无向图的关联矩阵，列出它的各边，并且给出每条边出现的次数.

(7)给定正整数 n，通过产生图的相邻矩阵来生成无向图，使得以相等的概率来生成所有的简单图.

(8)给定正整数 n，通过产生图的相邻矩阵来生成有向图，使得以相等的概率来生成所有的有向图.

26. 编程完成下列练习.

(1)显示带 6 个顶点的所有非同构的简单无向图.

(2)显示带 4 个顶点的所有有向图.

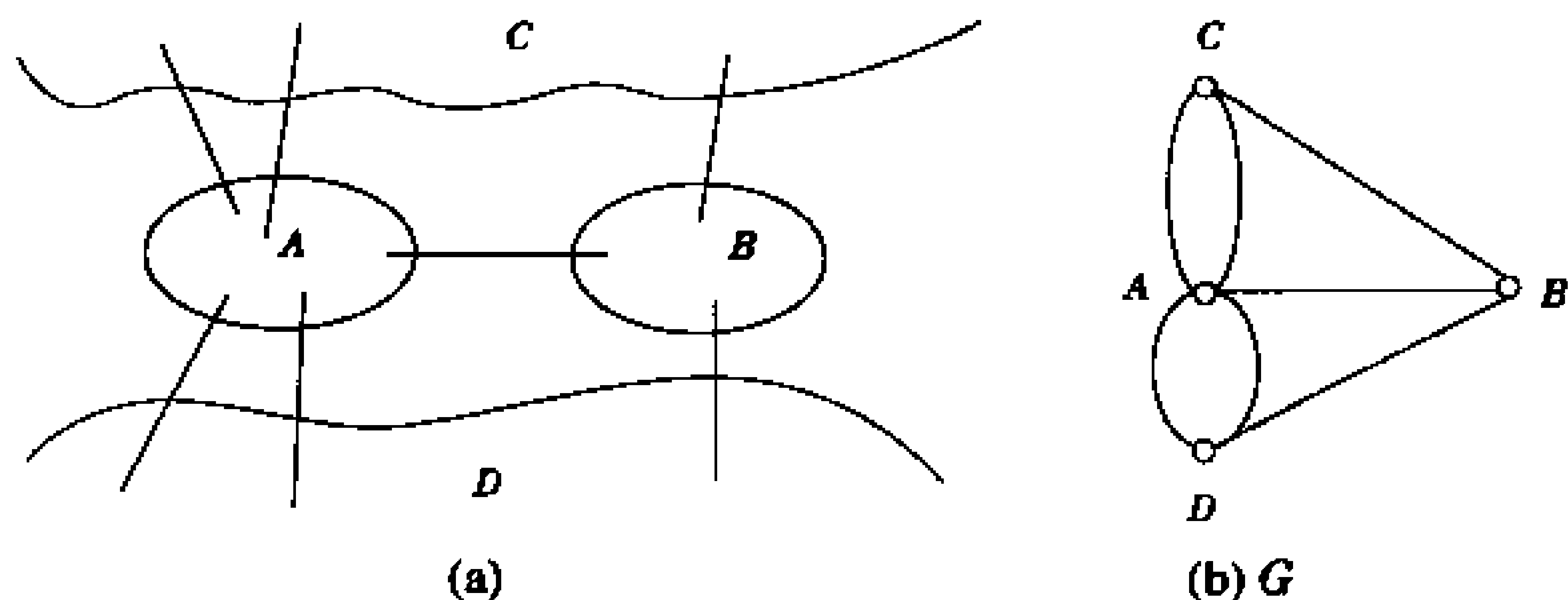
(3)随机地生成 10 个不同的简单图，每个带 20 个顶点，使得每个这样的图都是以相等的概率来生成的.

27. 描述判定图是否为二分图的算法.

第二章　欧拉图与哈密顿图

几乎所有图论著作都要介绍哥尼斯堡(königsberg)(现属俄罗斯共和国)七桥问题，这是瑞士著名的天才数学家欧拉(Leonhard Euler，1707—1783)发表的图论历史上第一篇论文.普雷格尔(Pergel)河横穿哥尼斯堡城，在陆地与河心两个小岛之间架有七座桥(如图 2-1(a)所示).一个有趣的问题是：一个人能否从城里某个地方出发，经过每座桥一次且仅一次，然后回到原来出发的地方？

欧拉把这个问题抽象成如图 2-1(b)所示的图 G，其中 G 的结点 A、B、C、D 分别表示四块陆地，两块陆地之间的桥用对应点之间的一边表示.于是哥尼斯堡七桥问题就等价于图 G 中是否存在经过图 G 的每边一次且仅一次的闭迹(回)问题.欧拉通过研究，对上述问题给出了否定回答.



(a)　　(b) G

图 2-1

第一节 欧拉图

定义 1 给定无孤立结点的无向图 G,经过图 G 每边一次且仅一次的迹称为**欧拉路**.经过图 G 每边一次且仅一次的回称为**欧拉回路**.

定理 1 连通图 G 具有欧拉回路当且仅当它的每个顶点都有偶数度.

证明 ($\Rightarrow$)设连通无向图 G 的一条欧拉回路为C.设 C 从顶点 x_1 开始,接着是 x_1 关联的一条边,不妨设为$\{x_1,x_2\}$.边$\{x_1,x_2\}$为 x_1 贡献 1 度.这条回 C 每次经过一个顶点 x_i 就为该顶点 x_i 贡献 2 度,这是因为回 C 总是经过与 x_i 关联的一边进入 x_i,又经过与 x_i 关联的另一边离开 x_i.最后,回 C 在它开始的 x_1 结束,为 x_1 再贡献 1 度.因此,$\deg(x_1)$为偶数,这是因为回 C 开始时为 x_1 贡献 1 度,结束时为 x_1 贡献 1 度,每经过 x_1 一次贡献 2 度.其余各顶点的度数也为偶数,这是因为回 C 每经过一次,都为其贡献 2 度.

($\Leftarrow$)设 G 是连通无向图,作 G 的一条最长回C,并假设 C 不是欧拉回路,则 $\exists x_k\in V(C)$及关联 x_k 的边 $e=\{x_k,x_1'\}$,且 $e\notin C$;由于 $\deg(x_1')$为偶数,于是 $\exists e_1=\{x_1',x_2'\},\cdots$,依次类推,存在 $e_2=\{x_2',x_3'\},e_3=\{x_3',x_4'\},\cdots,e_n=\{x_n',x_k\}$,于是得到又一条回 C'.把 C 与C'组合在一起,如果恰是 G,则即得 G 的一条欧拉回路;否则,继续上面的过程,总可以组合出 G 的一条回,使其经过 G 中每边一次且仅一次.证毕

定理 2 连通图 G 具有欧拉路而无欧拉回路,当且仅当 G 恰有两个奇数度结点.

证明 ($\Rightarrow$)设连通图 G 从顶点 a 到顶点 b 有欧拉路 C,但不

是欧拉回路. 欧拉路 C,每经过 G 中顶点 x_i(包括 a 和 b)一次,都为 x_i 贡献 2 度,而 C 的第一条边为 a 贡献 1 度,C 的最后一条边为 b 贡献 1 度. 因此,a 和 b 的度数均为奇数,其余结点度数均为偶数.

(⇐)设连通图 G 恰好有两个奇数度结点,不妨设为 a 和 b. 在图 G 中添加一条边 $e=\{a,b\}$ 得 G',则 G' 的每个结点的度数均为偶数,因而 G' 中存在欧拉回路,故得证 G 中一定存在欧拉路.

现在可以解决哥尼斯保七桥问题,如图 2-1(b)所示图 G 中,由于 $d(A)=5, d(B)=3, d(c)=3, d(D)=3$,因此 G 没有欧拉回路,也没有欧拉路.

定义 2 给定有向图 D,经过 D 中每边一次且仅一次的有向迹称为**有向欧拉路**. 经过 D 中每边一次且仅一次的有向闭迹(回)称为**有向欧拉回路**.

定理 3 具有弱连通性的有向图 G 具有有向欧拉回路,当且仅当 G 的每个结点的入度等于出度. 具有弱连通性的有向图 G 具有有向欧拉路,当且仅当在 G 中,一个结点的入度比出度大 1,另一个结点的入度比出度小 1,而其余每个结点的入度等于出度.

定义 3 含欧拉回路的无向连通图与含有向欧拉回路的弱连通有向图,统称为**欧拉图**.

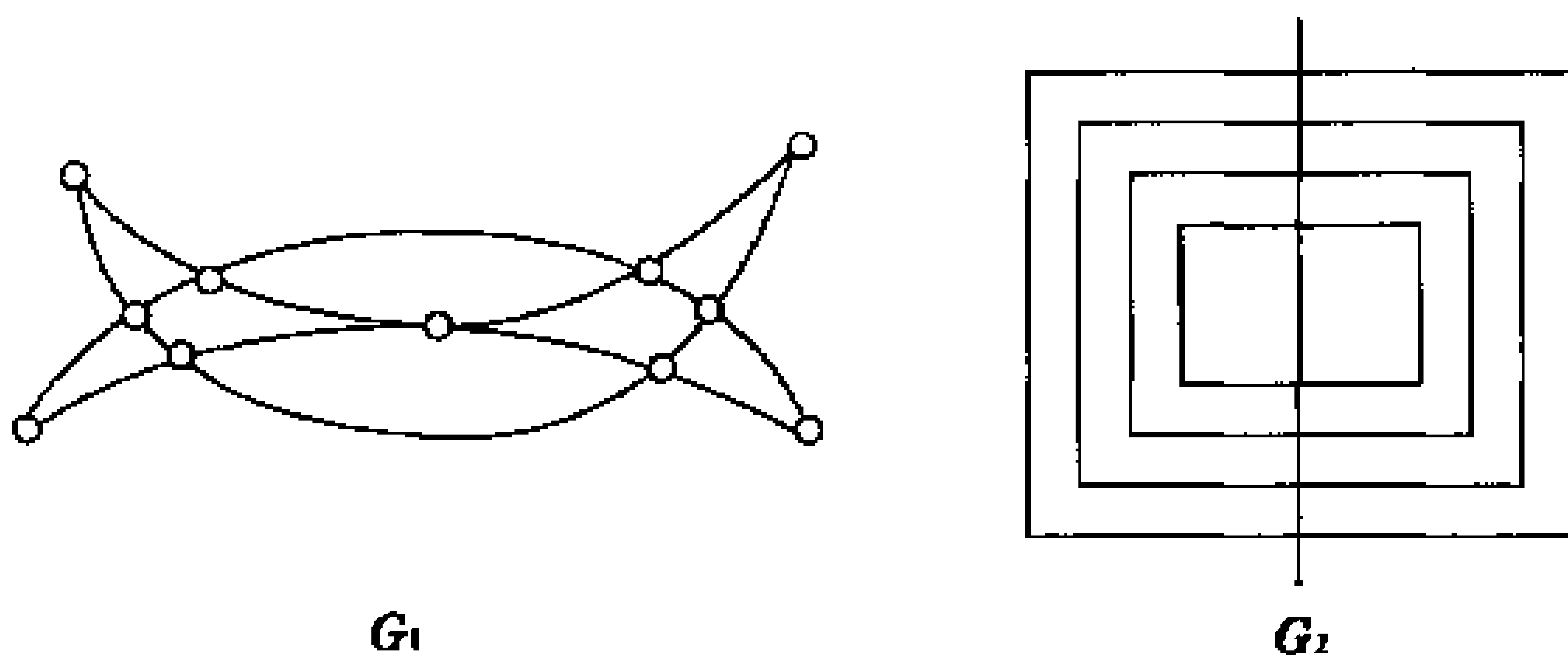


图 2-2

与哥尼斯堡七桥问题非常相似的有趣问题是中国古代一笔画问题，即要求用笔连续移动，不离开纸面并且不重复地画出图形.例如，能否一笔画出如图 2-2 所示图 G_1 和 G_2?

例 1 一个编码盘等分成 2^4 等份，每份分别用绝缘体和导体组成，可表示为 0 和 1 两种状态，其中 a、b、c、d 四个位置的扇面组成一个二进制输出，如图 2-3 所示，试问这 16 个二进制数列应如何排列，使圆盘沿顺时针每旋转一部分，恰好组成 0000 到 1111 的 16 组四位二进制输出，同时旋转一周又返回到 0000 状态?

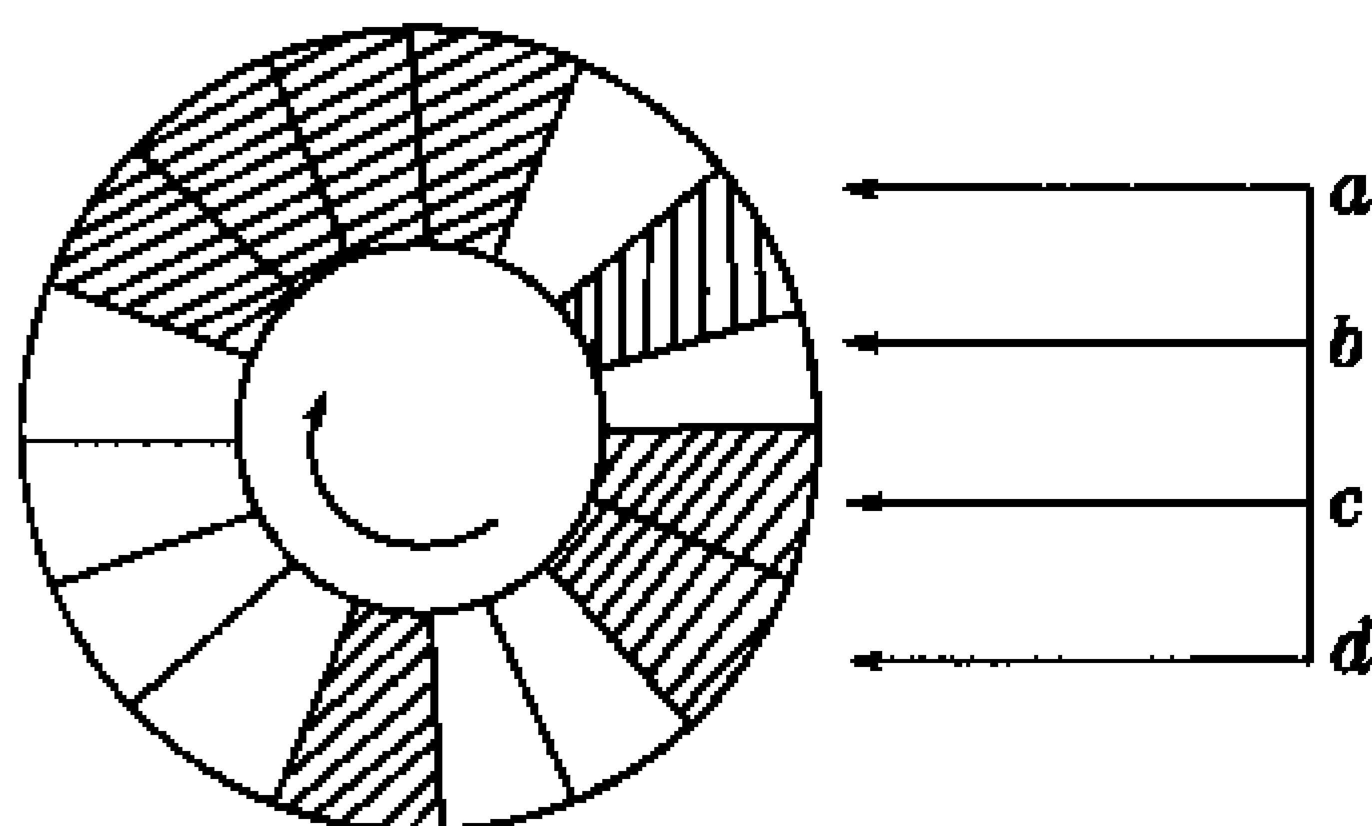


图 2-3

解 定义一个八个结点的有向图 $G=\langle V,E,\varphi\rangle$，如图 2-4 所示，其中

$$V=\{x_1x_2x_3\mid x_i\in\{0,1\},i=1,2,3\},$$
$$E=\{x_1x_2x_3x_4\mid x_i\in\{0,1\},i=1,2,3,4\},$$
$$\varphi:E\rightarrow V\times V,\text{且}$$
$$\varphi(x_{i-1}x_ix_{i+1}0)=\langle x_{i-1}x_ix_{i+1},x_ix_{i+1}0\rangle,$$
$$\varphi(x_{i-1}x_ix_{i+1}1)=\langle x_{i-1}x_ix_{i+1},x_ix_{i+1}1\rangle.$$

由于这样定义的图 G 中，每个结点的入度与出度都相等，因此图 G 中存在欧拉有向回路，如($e_0e_1e_2e_4e_9e_3e_6e_{13}e_{10}e_5e_{11}e_7e_{15}e_{14}e_{12}e_8$). 根据邻接边的定义规则，这 16 个二进制数可写成对应的二进制数序列 0000100110101111. 把这个序列排成环状，即与

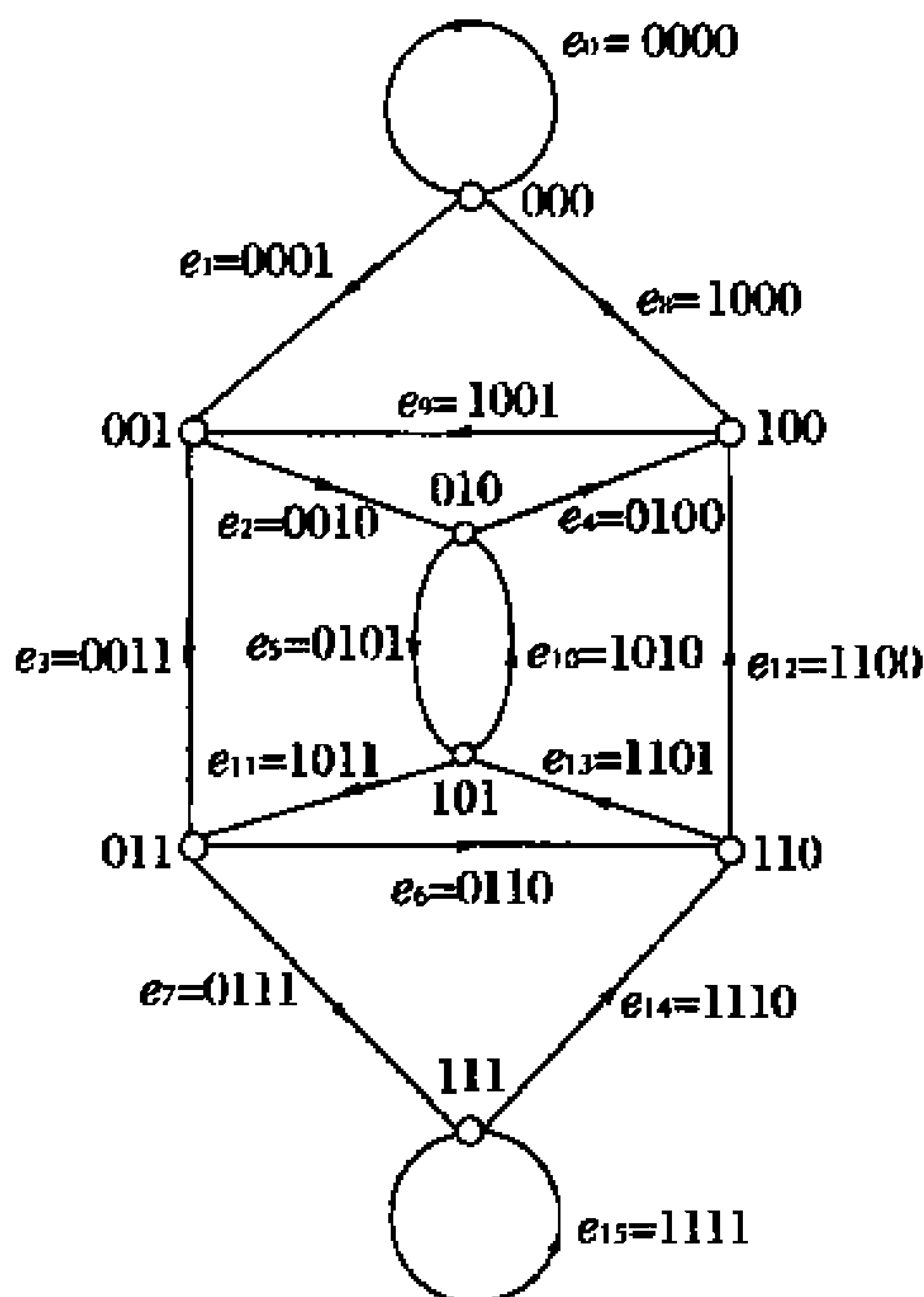


图 2-4

所求的鼓轮相对应,如图 2-3 所示.

上面的例子, 我们可以把它推广到鼓轮具有 n 个触点的情况.为此只要构造 2^{n-1}个结点的有向图,设每个结点标记为 $n-1$ 位二进制数,从结点 $x_1x_2\cdots x_{n-1}$出发,有一条终点为 $x_2x_3\cdots x_{n-1}0$ 的边,该边记为 $x_1x_2x_3\cdots x_{n-1}0$;还有一条终点为 $x_2x_3\cdots x_{n-1}1$ 的边,该边记为 $x_1x_2x_3\cdots x_{n-1}1$.这样构造的有向图为**德布鲁英(de Bruijn)图** $B(k,n)$的特例,即 $B(2,n)$.由于 $B(2,n)$的每个结点的入度等于出度,故 $B(2,n)$中存在有向欧拉回路,在该有向欧拉回路中,取每条边标号第一个数字构成的二进制序列即为**德布鲁英(de Bruijn)序列** $B(k,n)$的特例.

de Bruijn 序列在编码及计算机磁鼓设计中非常有用，有兴趣的同学可参考相应教材.

我们很容易判断一个图是否为 Euler 图，下面我们介绍一种求 Euler 图的 Euler 回路的 Fleury 算法.

Fleury 算法

(1)任意选取一个顶点 v_0，置 $W_0=v_0$；

(2)假定迹(若是有向图，则是有向迹) $W_i=v_0e_1v_1\cdots e_iv_i$ 已经选出，那么用下列方法从 $E(G)-\{e_1,e_2,\cdots,e_i\}$ 中选取 e_{i+1}；

(a) e_{i+1} 与 v_i 关联(若是有向图，e_{i+1} 以 v_i 为起点)；

(b)除非没有别的边可选择，e_{i+1} 不是 $G_i=G-\{e_1,e_2,\cdots,e_i\}$ 的割边.

(3)当(2)不能执行时，停止. 否则让 $i+1\to i$，转(2).

显然，Fleury 算法作出了 G 的一条迹.

定理 4 若 G 是 Euler 图，则 Fleury 算法终止时得到的 G 的迹是 Euler 回路.

证明 不妨设 G 是无向图，若 G 是有向图，证法类似.

设 G 是 Euler 图，$W_n=v_0e_1v_1e_2\cdots e_nv_n$ 是 Fleury 算法终止时得到的迹.

先证 W_n 是 G 的一条闭迹.

由于 $G_n=G-\{e_1,e_2,\cdots,e_n\}$，则在 G_n 中结点 v_n 的度数为 0，否则 G_n 中存在边 e_{n+1} 与 v_n 关联，因而算法到第 n 步时还可继续进行，矛盾. 如果 $v_n\neq v_0$，则在 W_n 中与 v_n 关联的边数为奇数，即 $d_{W_n}(v_n)$ 为奇数. 而 $d_{W_n}(v_n)+d_{G_n}(v_n)=d_G(v_n)$，这与 G 是 Euler 图矛盾，故 $v_n=v_0$. 因此 W_n 是 G 的一条闭迹.

再证 W_n 是 G 的 Euler 回路，即证明 $E(G)=E(W_n)$.

假设 W_n 不是 G 的 Euler 回路，则 $G_n=G-\{e_1,e_2,\cdots e_n\}$ 含有非平凡连通分支. 因为 G 是 Euler 图，而 W_n 是闭迹，因此 G_n

中没有奇度数结点，设 G'_n 是 G_n 的一个非平凡连通分支，显然 G'_n 是 Euler 图. 设 C'_n 就是 G'_n 的一条 Euler 回路.

由于 G 连通，可推得 W_n 与 C'_n 有公共顶点(否则 C'_n 就是 G 的一个连通分支，与 G 连通矛盾)，设 v_m 是沿着 W_n 而又在 C'_n 中下标最大的一个点(因为 v_n 不在 C'_n 中，故这种顶点必存在，并且 $v_n \neq v_m$). 显然 e_{m-1} 就是 G_m 的一条割边.

设 e 是 C'_n 中与 v_m 关联的一条边，由于 G'_n 是 Euler 图，因而 G'_n 没有割边，所以 e 不是 G'_n 的割边，因而也不是 G_n 的割边. 另一方面，由于 $m < n$，G_n 是 G_m 的生成子图，所以 e 也不可能是 G_m 的割边.

由于 Fleury 算法构造 W_m 的过程是：尽可能不取割边，但在构作 W_{m+1} 时没有取非割边 e(与 v_m 关联)而取了割边 e_{m+1}(也与 v_m 关联). 矛看. 故 $E(G_n) = \varnothing$，即 W_n 是包含 G 的所有边的闭迹. 证毕

不难看出，Fleury 算法的时间复杂度为 $O(|E(G)|)$.

第二节 哈密顿图

与“七桥问题”同样有趣的是“周游世界”问题. 这是著名的爱尔兰数学家哈密顿(William Rowan Hamilton, 1805—1865)在 1857 提出的一个游戏：能不能在图 2-5 所示的十二面体图中找出一条圈，使它经过图中每个结点一次且仅一次?

定义 1 给定无向图 G，若存在一条路经过图 G 的每个结点一次且仅一次，这条路称作**哈密顿路**. 若存在一条闭路经过图 G 的每个结点一次且仅一次，这条闭路称作**哈密顿回路**.

定义 2 给定有向图 D，经过 D 中每个结点一次且仅一次的有向路称作**哈密顿有向路**. 经过 D 中每个结点一次且仅一次的有

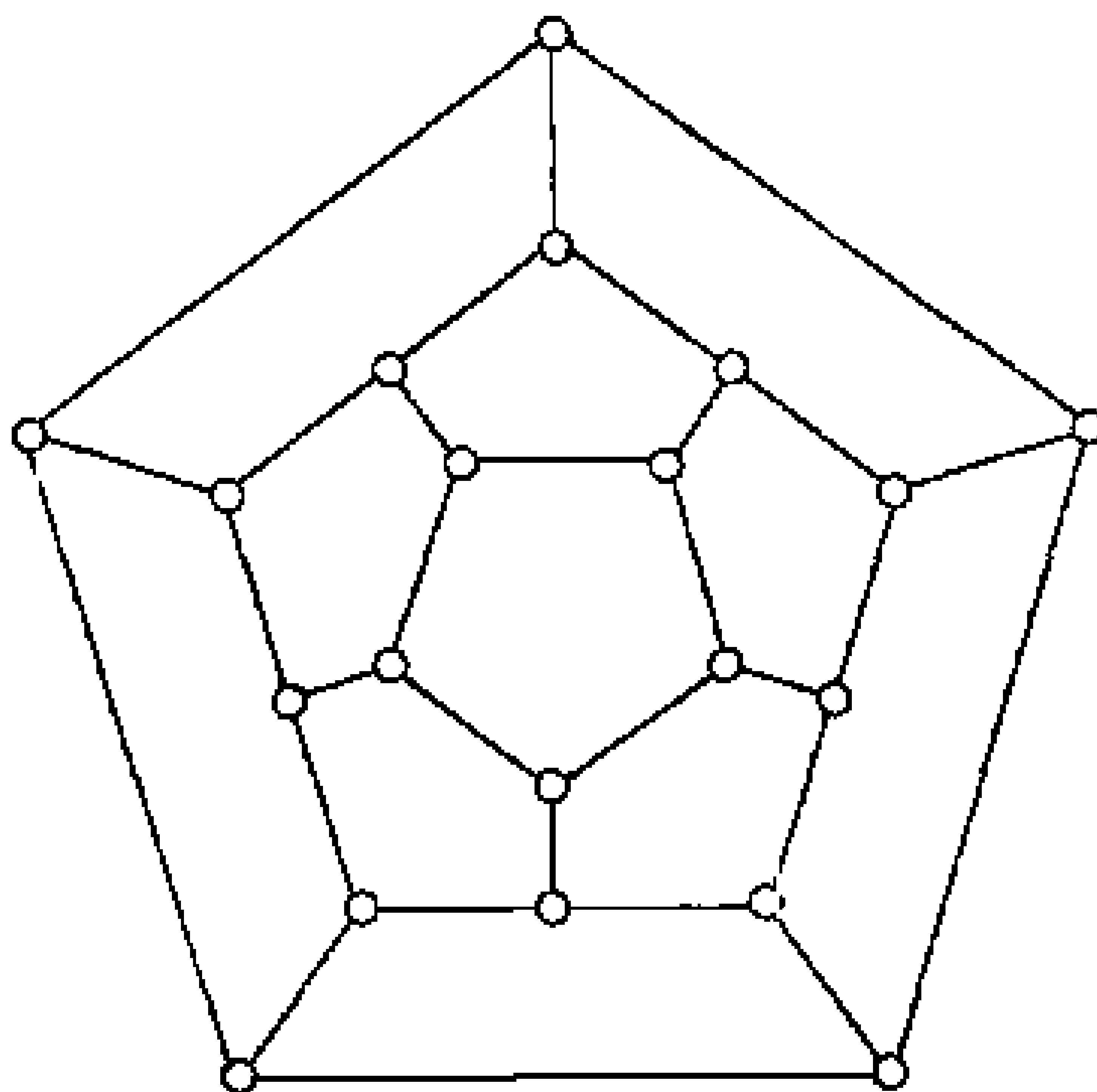


图 2-5

向闭路称作**哈密顿有向回路**.

定义 3 具有哈密顿回路的无向图,与具有哈密顿有向回路的有向图,统称为**哈密顿图**.

例 1 对于完全图 $K_n(n\geqslant3)$,由于 K_n 中任意两个顶点之间都有边,从 K_n 的某一顶点开始,总可以遍历其余结点后,再回到该结点,因而 $K_n(n\geqslant3)$是哈密顿图.

哈密顿问题,即判断一个给定的图是否是哈密顿图的问题,是图论中尚未解决的难题之一.本节介绍若干必要条件和充分条件.

定理 1 设任意 $n(n\geqslant3)$阶图 G,对所有不同非邻接顶点 x 和 y,若 $\deg(x)+\deg(y)\geqslant n$,则 G 是哈密顿图.

证明 仅就 G 是无向图加以证明.假设定理不成立.则存在一个阶为 $n(n\geqslant3)$,满足定理条件且边数最多的非哈密顿图,即 G 是一个非哈密顿图且对 G 的任何两个非邻接顶点 x_1 和 x_2,图 G + 边$\{x_1,x_2\}$是哈密顿图.因为 $n\geqslant3$,所以 G 不是完全图.

设 u 和 v 是 G 的两个不邻接顶点.因此 G + 边$\{u,v\}$是哈密

顿图且 $G+$边$\{u,v\}$的哈密顿回路一定包含边$\{u,v\}$.由此可得,在 G 中存在一条uv 路$T=u_1u_2\cdots u_n(u=u_1,v=u_n)$包含 G 中每个顶点.

如果$\{u_1,u_i\}\in E(G)(2\leqslant i\leqslant n)$,那么$\{u_{i-1},u_n\}\notin E(G)$.否则,$u_1u_iu_{i+1}\cdots u_nu_{i-1}u_{i-2}\cdots u_1$ 是 G 的一个哈密顿回路,所以,对$\{u_2,u_3,\cdots,u_{n-1}\}$中每一个邻接到 u_1 的顶点存在一个$\{u_1,u_2,\cdots,u_{n-1}\}$中与 u_n 不邻接的顶点,因此,$\deg(u_n)\leqslant n-1-\deg(u_1)$,从而 $\deg(u)+\deg(v)\leqslant n-1$.与定理的条件矛盾,所以 G 是哈密顿图.证毕.

如果 G 是哈密顿图,显然 $G+$边$\{u,v\}$也是哈密顿图.这里 u 和 v 是不同的不邻接点.反之,假设 G 是一个 n 阶图,u 和 v 是不邻接顶点,$G+$边$\{u,v\}$是哈密顿图,且 $\deg(u)+\deg(v)\geqslant n$,则 G 是哈密顿图,否则,象定理 1 的证明中那样,得到矛盾$\deg(u)+\deg(v)\leqslant n-1$.由此得到下面结果.

定理 2　设 u 和 v 是 n 阶图 G 的不同非邻接点,且 $\deg(u)+\deg(v)\geqslant n$,则 $G+$边$\{u,v\}$是哈密顿图当且仅当 G 是哈密顿图.

定理 2 暗示了图的一个新概念,即图的闭包.

定义 4　给定 n 阶图G,若将图 G 中度数之和至少是n 的非邻接点用一条边连结起来得图G',对图 G'重复上述过程,直到不再有这样的结点对存在为止,所得到的图,称为是原图 G 的**闭包**(**closure**),记作 $C(G)$.

图 2-6 说明了图 G 闭包的构造过程.

定理 3　一个图是哈密顿图,当且仅当它的闭包是哈密顿图.

定理 4　设 G 是阶至少为 3 的图,如果 G 的闭包是完全图,那么 G 是哈密顿图.

定理 5　如果 G 是一个$n(n\geqslant3)$阶任意图,且对 G 的每个顶点 x,都有 $\deg(x)\geqslant n/2$,则 G 是哈密顿图.

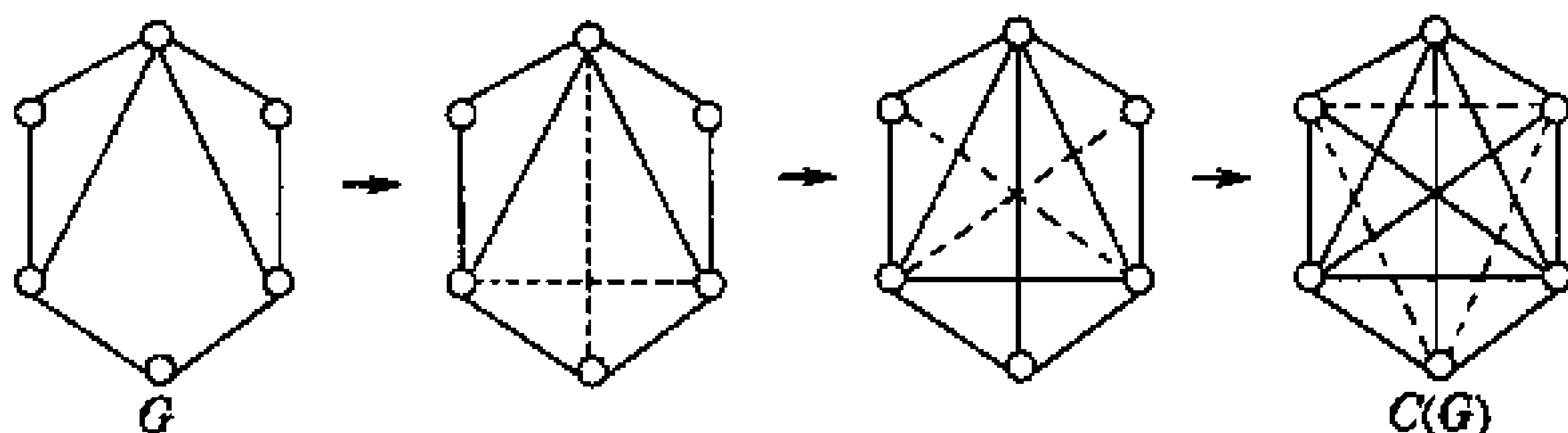


图 2-6

由哈密顿图的定义知，哈密顿有向图必是强连通的，哈密顿无向图必无割点. 下面再介绍一个必要条件.

定理 6 若 G 是一个哈密顿图，则对于 $V(G)$ 的每个非空真子集 S，$W(G-S)\leqslant|S|$. 其中 $W(G-S)$ 为 $G-S$ 的分支数.

证明 设 C 是 G 的一个哈密顿回路，则对于 $V(G)$ 的任意一个非空真子集 S，均成立 $W(C-S)\leqslant|S|$. 由于 $C-S$ 为 $G-S$ 的一个生成子图，因而 $W(G-S)\leqslant W(C-S)$，于是 $W(G-S)\leqslant|S|$.

该条件只是必要条件，如图 2-7 所示的 petersen 图，满足 $W(G-S)\leqslant|S|$，但它是非哈密顿图.

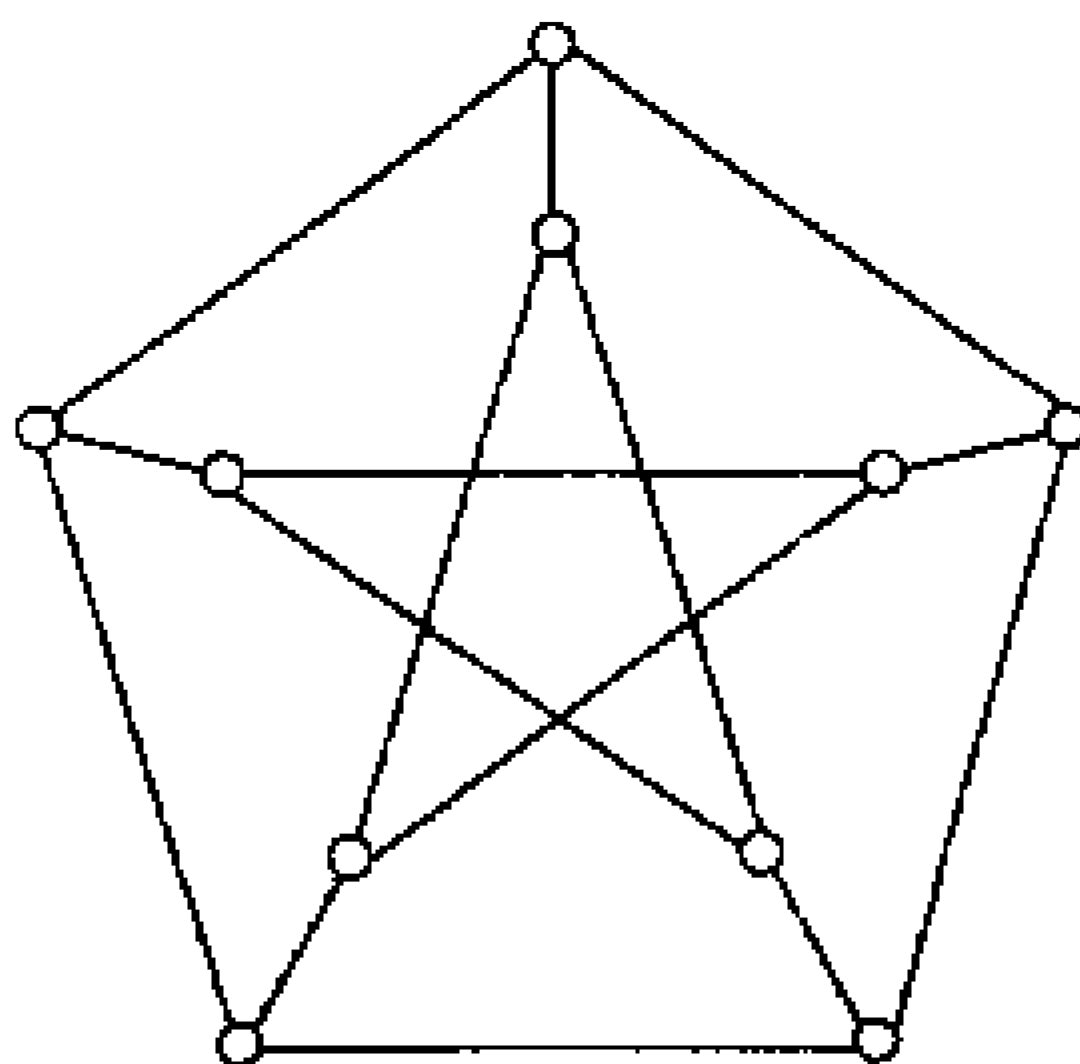


图 2-7 petersen 图

例 1 11 个学生要共进晚餐，他们将坐成一个圆桌，计划要求

每次晚餐上，每个学生有完全不同的邻座．这样能共进晚餐几次？

解 将每个学生看作图的顶点，学生的邻座关系作为图的边，学生每次进餐的就座方式对应一个哈密顿回路．两次进餐中，每个学生有完全不同的邻座对应着两个没有公共边的哈密顿回路．因为每个学生都可以与其余学生邻座，所以问题转化为在图 K_{11} 中找出所有没有公共边的哈密顿回路的个数．

K_{11} 中共有 $11\times(11-1)/2=55$ 条边，而 K_{11} 中每条哈密顿回路的长度为 11，因此 K_{11} 中最多有 $\frac{55}{11}=5$ 条没有公共边的哈密顿回路．另一方面，我们可以构造 5 条没有公共边的哈密顿回路，作图如下：

设第一条哈密顿回路为 $(1,2,3,\cdots,11,1)$，把 1 固定在圆心，其余点固定在圆周上，如图 2-8(a)所示，然后将此图的顶点旋转 $360°/10, 2\times360°/10, 3\times360°/10, 4\times360°/10$，就得到另外 4 个图（如图 2-8($b$)～($e$)所示）．所以学生可以进 5 次晚餐，进餐座位如下：

1,2,3,4,5,6,7,8,9,10,11,1

1,4,2,6,3,8,5,10,7,11,9,1

1,6,4,8,2,10,3,11,5,9,7,1

1,8,6,10,4,11,2,9,3,7,5,1

1,10,8,11,6,9,4,7,2,5,3,1

例 2 有 n 个人，任意两个人合起来认识其余的 $n-2$ 个人，证明：当 $n\geqslant4$ 时，这 n 个人能站成一圈，使每一个人的两旁站着自己认识的人．

证明 构造简单无向图 $G=\langle V,E\rangle$，其中 V 中的 n 个结点表示这 n 个人，G 中的边表示他们间的认识关系，若两个结点间有一条边，当且仅当对应的人互相认识．

任取 $u,v\in V(G)$，显然 $d(u)+d(v)\geqslant n-2$，且其余 $n-2$

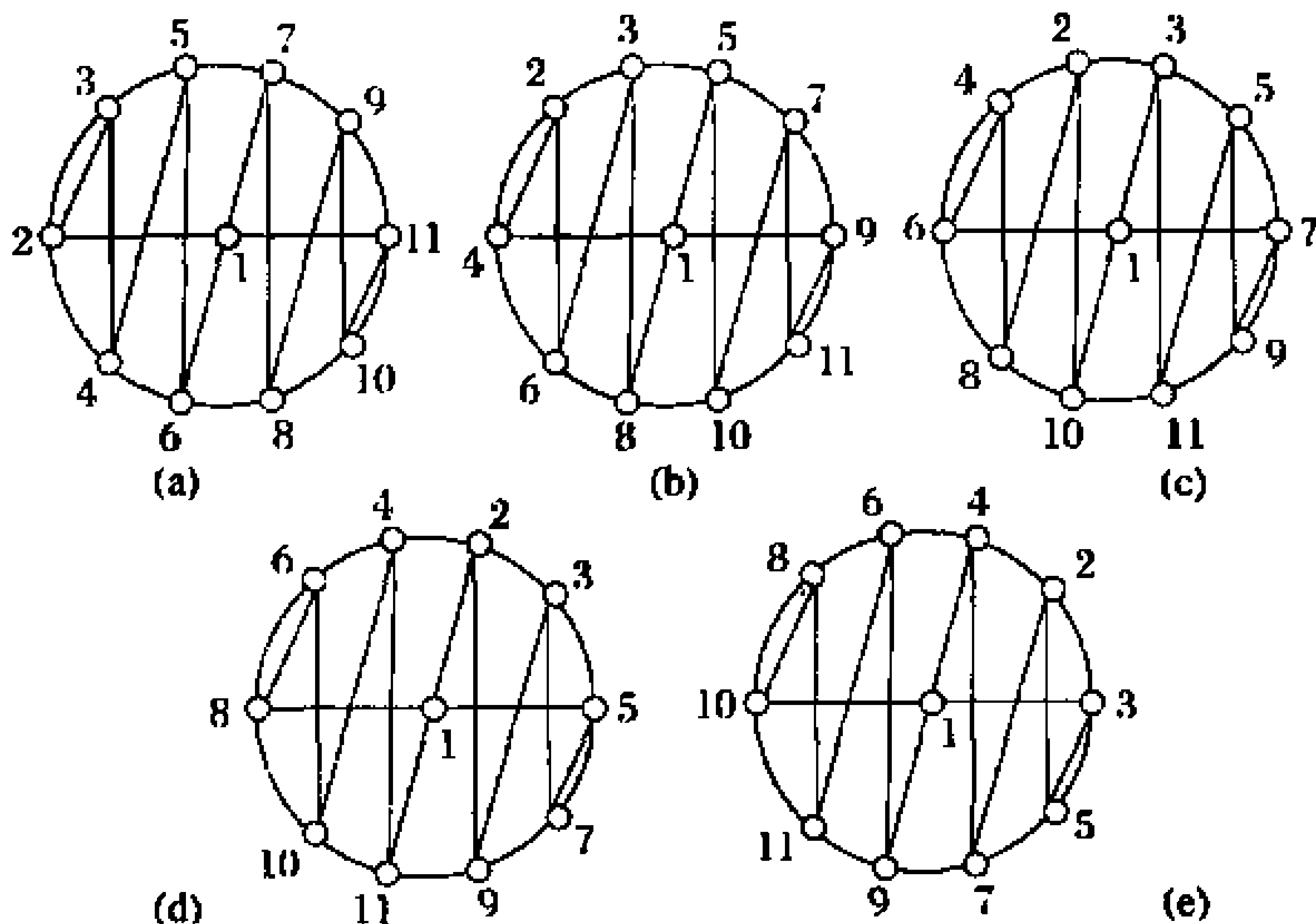


图 2-8

个结点必与 u 或 v 邻接.

(1)若 u 与 v 相邻,则 $d(u)+d(v)\geqslant 2+n-2=n$

(2)若 u 与 v 不相邻,如果 $d(u)+d(v)=n-2$.则 $V-\{u,v\}$中恰有 $n-2$ 个结点($n\geqslant 4$,故 $V-\{u,v\}\neq\varnothing$),其中每一个结点只能与 u,v 中的一个结点相邻.不妨设 $a\in V-\{u,v\}$,且 a 与 u 相邻,u 与 v 不相邻,此时对于结点 a 与 u 来说,都不与 v 相邻,这与已知矛盾,因此 $d(u)+d(v)>n-2$,即 $d(u)+d(v)\geqslant n-1$.若 $d(u)+d(v)=n-1$,由于 $n\geqslant 4$,在结点集 $V-\{u,v\}$ 中至少有两个结点 u 和b,其中 a 与 u 和 v 都相邻,而 b 只与 u 和 v 中的一个相邻,不妨设 b 与 u 相邻,这样 b 与 u 和 v 都不相邻,显然与已知矛盾,因此 $d(u)+d(v)>n-1$,即 $d(u)+d(v)\geqslant n$.

综上所述,对于 $\forall u,v\in V(G)$,总有 $d(u)+d(v)\geqslant n$,因此

G 中存在一条哈密顿回路，于是这 n 个人能站成一圈，使每一个人的两旁站着自己认识的人.

第三节　并行运算图论模型与格雷码

传统的计算机一般称为**串行计算机**，即执行程序是一次完成一个操作.因此，为解决问题而写的算法都设计成一次执行一步，这样的算法称为**串行的**.几乎所有书本描述的算法都是串行的.不过，像气象模拟，医学图像及密码分析这样的许多高强度计算问题，即使在超计算机上，也不能通过串行操作在合理时间范围内解决；而且，对计算机执行基本操作有多快来说，还存在物理的限制.所以，总是有问题不能用串行操作在合理长度的时间之内解决.同时，随着硬件价格的下降，使得制造并行计算机成为可行.**并行计算机**就是使用多个处理器实现在同一时间执行多条指令.利用包含许多自带内存的单处理器的计算机来帮助克服串行计算机的局限性.**并行算法**把问题分成可并发解决的若干子问题，其单个指令流控制着算法的执行，包括把子问题送到不同的处理器，以及把子问题的输入和输出定向到适当的处理器.

采用并行处理，一个处理器需要另一个处理器产生的输出.因此，处理器需要互联.用图来描述带有多重处理器的计算机里处理器的互联网络，是一种十分便利的方法.我们在第一章中所介绍的 n 维立方体(超立方体) Q_n 是一个重要类型的互联网络。在 n 维立方体 Q_n 中，有 2^n ($n\geqslant1$)个处理器，每个处理器有自己的内存.在一个单元时间内，n 维立方体中的每个处理器同时处理一条指令，然后与之相邻的处理器进行通信.若一个处理器要与一个不相邻的处理器通信，第一个处理器要发送一些信息，这包括路径信息及接受者的最终目的地.当然需花费数个时间段.

许多计算机已经根据 n 维立方体 Q_n 的模型制造和运行.我

们将证明 n 维立方体 Q_n 中存在哈密顿回路,为此我们先介绍格雷码.格雷码是以弗兰克·格雷(Frank Gray)的名字命名的,在20世纪40年代,他为了把传送数字信号过程中的错误的影响降到最低而发现了他们.

定义1 n 阶格雷码是序列 $s_1, s_2, \cdots, s_{2^n}$,其中每个 s_i 是一个 n 位的二进制串,满足

(1)每个 n 位二进制串都出现在序列中;

(2)s_i 与 s_{i+1} 只有一位不同,$i=1,2,\cdots,2^n-1$;

(3)s_{2^n} 和 s_1 只有一位不同.

格雷码已广泛地应用于各个领域,如模拟信息和数字信息的转换等.下面介绍如何为每个正整数构造格雷码.

定理1 令 G_1 表示序列0,1.由 G_{n-1} 根据以下规则来定义 G_n:

(1)令 G_{n-1}^R 表示序列 G_{n-1} 的逆序;

(2)令 G_{n-1}' 表示在 G_{n-1} 的每个成员前加0所得序列;

(3)令 G_{n-1}'' 表示在 G_{n-1}^R 的每个成员前加1所得序列;

(4)令 G_n 为由 G_{n-1}' 后加上 G_{n-1}'' 组成的序列.

则 G_n 是正整数 n 的格雷码.

证明 $n=1$ 时,定理显然成立.

假设 G_{n-1} 是格雷码.G_{n-1}' 中的每个串以0开始,因此 G_{n-1}' 中的任意连续串之间的不同必定由 G_{n-1} 中对应的串的不同所决定.但是 G_{n-1} 是格雷码,G_{n-1} 中每个连续的串对之间只有一位不同.因此,G_{n-1}' 中每个连续的串对也只有一位不同.类似的,G_{n-1}'' 中每个连续的串对也只有一位不同.

令 α 表示 G_{n-1}' 中的最后一个串,令 β 表示 G_{n-1}'' 中的第一个串.若删去 α、β 中的第一位,则所得的串是相同的.由于 α 的第一

位为 0，β 的第一位为 1，所以 G'_{n-1} 的最后一个串和 G''_{n-1} 的第一个串只有一位不同．类似的，G'_{n-1} 的第一个串和 G''_{n-1} 的最后一个串也只有一位不同．因此证得 G_n 为格雷码．

例 1 从 G_1 开始来构造 G_3．

G_1	0	1						
G_1^R	1	0						
G_1'	00	01						
G_1''	11	10						
G_2	00	01	11	10				
G_2^R	10	11	01	00				
G_2'	000	001	011	010				
G_2''	110	111	101	100				
G_3	000	001	011	010	110	111	101	100

推论 对于每个正整数 $n \geqslant 2$，n 维立方体都包含一个哈密顿回路．

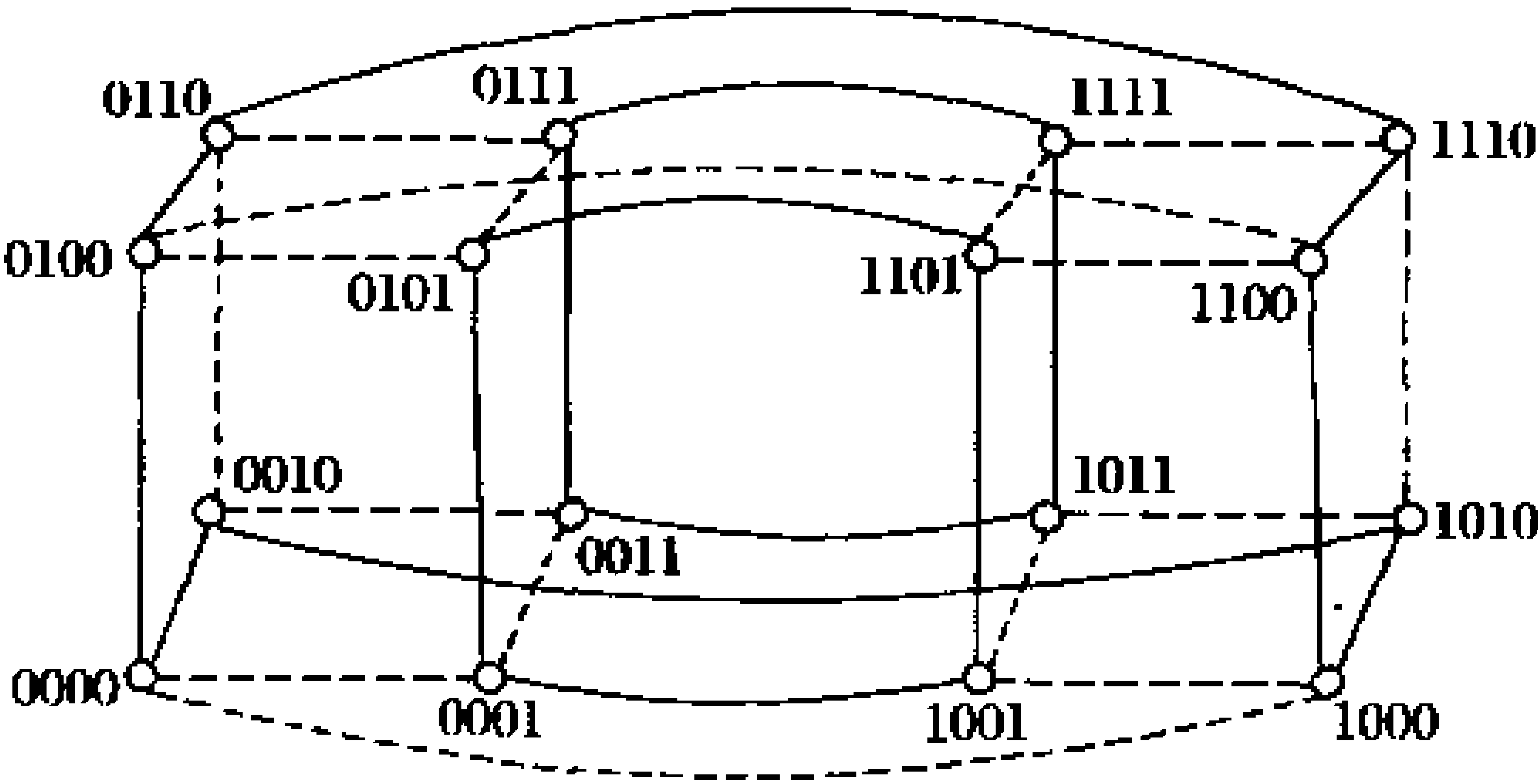


图 2-9

例如，Q_4 的一条哈密顿回路显示在图 2-9 中．

例 2 （1971 年波兰数学奥林匹克试题)证明任一个有限集合的全部子集可以这样排列顺序,使任何相邻的两个子集仅相差一个元素.

证明 设此有限集为 $A=\{a_1,a_2,\cdots,a_n\}$,我们用长为 n 的二进制数列对应它的 2^n 个子集:当且仅当子集中含有 A 的第 i 个元素时,数列的第 i 个数码为 1,否则为 0. 以这 2^n 个数列为顶点,当且仅当两个数列仅一个同位数码相异时,此两顶点间连一边,得到一个图 G. 显然图 G 为 n 维立方体,而 n 维立方体是哈密顿图,按照一条哈密顿回路的顺序排列对应子集即可.

比如,图 2-9 所示为 4 维立方体,由格雷码得到它的一条哈密路为 0000 - 0001 - 0011 - 0010 - 0110 - 0111 - 0101 - 0100 - 1100 - 1101 - 1111 - 1110 - 1010 - 1011 - 1001 - 1000.

因此,对于集合 $A=\{a_1,a_2,a_3,a_4\}$,对应子集的一种排列为:$\varnothing$,$\{a_4\}$,$\{a_3,a_4\}$,$\{a_3\}$,$\{a_2,a_3\}$,$\{a_2,a_3,a_4\}$,$\{a_2,a_4\}$,$\{a_2\}$,$\{a_1,a_2\}$,$\{a_1,a_2,a_4\}$,$\{a_1,a_2,a_3,a_4\}$,$\{a_1,a_2,a_3\}$,$\{a_1,a_3\}$,$\{a_1,a_3,a_4\}$,$\{a_1,a_4\}$,$\{a_1\}$.

第四节　算法的时间复杂性

一个算法是一个有限指令的集合. 这些指令确定了解决某一问题的运算或操作序列. 算法具有以下性质:

有限性 任何算法都会在有限条指令执行完毕后结束.

确定性 算法的每一步有精确的定义.

输入 有零个或多个输入,且输入取自特定的对象集合.

输出 有一个或多个输出,与输入有某种特定关系.

唯一性 每一步执行后所得到的中间结果是唯一的,且仅依赖于输入和先前步骤的结果.

通用性 算法适用于一类输入.

仅具备后五个特性,称其为**计算过程**.如求$\sqrt{2}$的值是一个计算过程.

问题是要求回答的提问.通常有几个参数或自变量,它们的值是特定的,取自问题的定义域.**问题的描述**即对其参数进行描述,指出其解是满足什么性质的命题.给问题的全体参数都指定了确定的值,便得到问题的一个**实例**.例如,"$x+y=?$"是加法问题,而"$3+5=?$"是加法实例.当然,一个问题一般可包含无穷多个实例.

如果 A 算法对问题 P 的每个实例都给出正确答案,称 A 是**问题 P 的算法**.若存在一个算法解答问题 P,称问题 **P 算法可解**.

一个计算机程序即使是按照正确的算法编写的,对于一些输入来说也许是毫无用处的,因为这个程序执行的时间太长,或者程序的数据、变量等占用的存储空间太大.**算法分析**是指通过分析得到算法所需时间和空间的估计量.**算法的复杂度**是指执行算法所需的时间和空间的量.本节讲解如何估计算法执行所需时间的问题.

定义 1 令 f 和 g 为从整数集合或实数集合到实数集合的函数,如果存在常数 c 和 k,使得只要 $x>k$,就有

$$|f(x)|\leqslant c|g(x)|$$

则称 $f(x)$是 $O(g(x))$,记作 $f(x)=O(g(x))$,读作 $f(x)$是大 $Og(x)$.

例如,$f(x)=x^2+2x+1$ 是 $O(x^2)$.因为只要 $x>1$,就有 $0\leqslant x^2+2x+1\leqslant x^2+2x^2+x^2=4x^2$,即取 $k=1,c=4$.若 $f(x)=O(g(x))$,而对足够大的 x,$|h(x)|>|g(x)|$,显然当 $x>k$ 时,$|f(x)|\leqslant c|g(x)|\leqslant c|h(x)|$.因此,在使用大 O 符号时,在 $f(x)=O(g(x))$这一关系中函数 g 总是选得尽可能小(有时选自某个参考函数集合,如形为 x^n 的函数,其中 n 为正整数).

定理 1 令 $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$,其中

$a_0, a_1, \cdots, a_{n-1}, a_n$ 为实数,则 $f(x)=O(x^n)$.

证明 如果 $x>1$,我们有

$$
\begin{aligned}
|f(x)| &= |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0| \\
&\leqslant |a_n| x^n + |a_{n-1}| x^{n-1} + \cdots + |a_1| x + |a_0| \\
&= x^n\left(|a_n| + |a_{n-1}|\cdot\frac{1}{x} + \cdots + |a_1|\cdot\frac{1}{x^{n-1}} + |a_0|\cdot\frac{1}{x^n}\right) \\
&\leqslant x^n(|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|)
\end{aligned}
$$

所以,取 $k=1, c=|a_n|+|a_{n-1}|+\cdots+|a_1|+|a_0|$ 时,有 $|f(x)| \leqslant c(x^n)$,即 $f(x)=O(x^n)$.

定理 2 设 $f_1(x)=O(g_1(x)), f_2(x)=O(g_2(x))$,则 $f_1(x)+f_2(x)=O(\max(g_1(x),g_2(x)))$.

证明 设有常数 c_1, c_2, k_1, k_2,使得

$x>k_1$ 时,$|f_1(x)| \leqslant c_1|g_1(x)|$;

$x>k_2$ 时,$|f_2(x)| \leqslant c_2|g_2(x)|$.

而当 $x>\max(k_1,k_2)$ 时,

$$
\begin{aligned}
|f_1(x)+f_2(x)| &\leqslant |f_1(x)| + |f_2(x)| \\
&\leqslant c_1|g_1(x)| + c_2|g_2(x)| \\
&\leqslant c_1|g(x)| + c_2|g(x)| \\
&= (c_1+c_2)|g(x)| \\
&= c|g(x)|
\end{aligned}
$$

其中 $c=c_1+c_2, g(x)=\max(|g_1(x)|,|g_2(x)|)$.

定理 3 设 $f_1(x)=O(g_1(x)), f_2(x)=O(g_2(x))$,则 $f_1(x)f_2(x)=O(g_1(x)g_2(x))$.

例 1 对于正整数 n,给出 $n!$ 和 $\log n!$ 的大 O 估计.

解
$$
\begin{aligned}
n! &= 1\times 2\times 3\times\cdots\times n \\
&\leqslant n\times n\times n\times\cdots\times n = n^n
\end{aligned}
$$

而 $\log n! \leqslant \log n^n = n\log n$

所以 $n! = O(n^n)$，$\log n! = O(n\log n)$

例 2 给出 $f(n)=3n\log n! + (n^2+3)\log n$ 的大 O 估计，其中 n 是一个正整数.

解 由于 $\log n! = O(n\log n)$，$3n = O(n)$，所以，$3n\log n! = O(n^2\log n)$.

而 $n^2+3 = O(n^2)$，故 $(n^2+3)\log n = O(n^2\log n)$，因此，$f(n) = O(n^2\log n)$

例 3 给出 $f(x)=(x+1)\log(x^2+1)+3x^2$ 的大 O 估计.

解 由 $x>1$ 时，$x^2+1\leqslant 2x^2$，而 $x>2$ 时，

$$\log(x^2+1)\leqslant\log 2x^2=\log 2+\log x^2=\log 2+2\log x\leqslant 3\log x$$

所以 $\log(x^2+1)=O(\log x)$，而 $x+1=O(x)$.

因此，$(x+1)\log(x^2+1)=O(x\log x)$

再由 $3x^2 = O(x^2)$，而且 $x>1$ 时，$x\log x\leqslant x^2$，所以 $f(x) = O(x^2)$.

大 O 符号常用于估计一个特定的计算机过程或算法解题时需要的操作步数. 做这些估计时常用的函数包括：$1, \log n, n, n\log n, n^2, 2n, n!$

用微积分可以证明，上列函数中每个函数都小于表中列在它后面的函数，即每个函数与它后面的函数的比在 n 无限增长时趋于 0. 图 2-10 显示了这些函数的图像，图中函数值的每个刻度都是它前面刻度的两倍.

怎样分析算法的效率呢？一种效率度量是计算机按此算法解题所花的时间，另一种度量是计算机实现这一算法需要多大内存，当然都是假定输入值的规模是一定的. 前者称为算法的**时间复杂性**，后者称为算法的**空间复杂性**. 空间复杂性与实现算法时使用的特定数据结构紧密相关，这将在有关课程中详细讨论.

假设是“标准”的计算机. 算法的时间复杂性的考虑可以在输入规模一定的情况下，用算法使用的基本操作（解问题的关键操

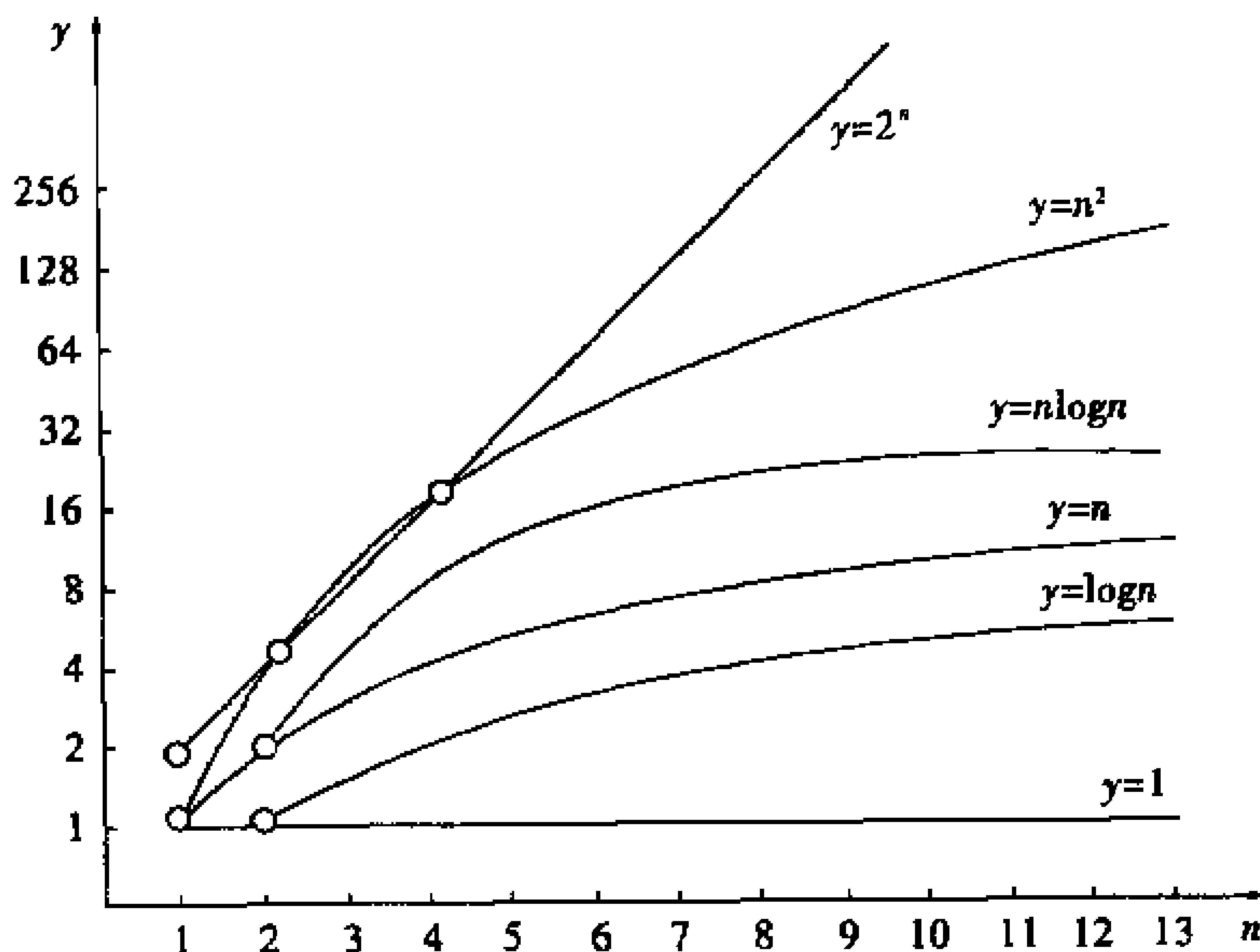


图 2-10

作,操作次数随输入规模增大而增加)次数来表示.而不用计算机实际运行的时间来表示,这是因为执行基本操作时,不同的计算机需要的时间不同,而且把所有运算分解成计算机使用的基本字位运算是相当复杂的.

在所有输入规模为 n 的情况下,可以找到执行算法所需的最短时间和最长时间,分别称为输入规模为 n 时的**最好时间复杂性**和**最坏时间复杂性**.**平均时间**是指在一组有限的规模为 n 的输入中执行算法所需的时间.平均时间复杂分析比最坏情况分析复杂得多.

表 2－1 中给出了描述算法时间复杂性的几个常用术语.

表 2－1

复杂性	术　语	复杂性	术　语
$O(1)$	常数复杂性	$O(n^k)$	多项式复杂性
$O(\log n)$	对数复杂性	$O(b^n), b>1$	指数复杂性
$O(n)$	线性复杂性	$O(n!)$	阶乘复杂性
$O(n\log n)$	$n\log n$ 复杂性		

能用具有多项式最坏情况复杂性的算法解决的问题称为**易处理的**，否则，称为**不易处理的**. 不过，如果在大 O 估计中的多项式次数高(如 100 次)，或多项式的次数特别大，算法可能会花特别长的时间来解题. 对于实际应用中出现的不易处理的问题，如果存在求近似解的快速算法，甚至还能保证近似解与精确解相差不太大，通常不求其精确解，而求其近似解.

易处理的问题属于 **P 问题**. 至今既没有找到多项式算法，又不能证明它不存在多项式算法，这类问题称为 **NP 问题**. 还有一类重要问题，只要其中任何一个问题能用一个多项式时间最坏情况算法来解，那么所有这些问题都能用多项式时间最坏情况算法解答，称其为 **NP 完全问题(non-deterministic polynomial complete problem)**，简称 **NPC 问题**.

随着科技的不断发展，计算机速度和内存空间获得巨大增长，再加上使用能做并行处理的算法，几年前认为无法解的问题现在可以当做日常事务处理了，这句话在以后仍然成立！

例 4　冒泡排序　它把一个列表这样排列成升序：相继地比较相邻的元素，若它们顺序不对，则交换它们. 试分析冒泡排序的最坏时间复杂度.

解　为完成冒泡排序，从表头开始执行基本操作，即交换一个较大元素与它后面的较小元素，对整个列表完全执行一遍. 让这个过程迭代，直到排序宣告完成为止. 如用表 2－2 显示用冒泡排序把 3,2,4,1,5 排成升序. 其中，(表示作一次交换. (表示处在正

确顺序的对. 5 表示处在正确顺序的数.

因为在第 k 遍使用 $n-k$ 次比较,所以在 n 个元素的列表的冒泡排序里使用的总比较次数是

$$(n-1)+(n-2)+\cdots+2+1=\frac{n(n-1)}{2}$$

必须注意,即使在某个中间步骤上这个列表变成完全排好顺序,这个算法仍然继续进行. 因此,冒泡排序算法有最坏时间复杂性 $O(n^2)$.

表 2-2

第 1 遍				第 2 遍			第 3 遍		第 4 遍
3	2	2	2	2	2	2	2	1	1
2	3	3	3	3	3	1	1	2	2
4	4	4	1	1	1	3	3	3	3
1	1	1	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5

第五节　最短路问题

边上有数的图称为**加权图(weighted graph)**. 若边 e 标记数 k,称边 e **的权(weight)**为 k. 在加权图中,**链(迹、路)的长度**为链(迹、路)上的所有边的权值的和. 在加权图中,我们经常需要找出两个指定点之间的**最短路**(如有最小长度的路),通常称其为**最短路问题(shortest path problem)**. 解决最短路问题存在几个不同的算法. 我们要介绍的是**迪克斯屈拉算法**,这是荷兰计算机科学教授 Edsger W. Dijkstra(1930 -)在 1959 年发现的一个算法. 他在

1972 年获得美国计算机协会授予的图灵奖，这是计算机科学中最具声望的奖项之一.

迪克斯屈拉算法是求出一个连通加权简单图中从结点 a 到 z 的最短路. 边 $\{i,j\}$ 的权 $w(i,j)>0$，且结点 x 的标号为 $L(x)$. 结束时，$L(z)$ 是从 a 到 z 的最短路的长度.

算法 迪克斯屈拉算法

procedure Dijkstra(G:所有权都为正数的加权连通简单图)

{G 带有顶点 $a=v_0, v_1, \cdots, v_n=z$ 和权 $w(v_i, v_j)$，若 $\{v_i, v_j\}$ 不是 G 中的边，则 $w(v_i, v_j)=\infty$}

for $i:=1$ **to** n

$L(v_i):=\infty$

$L(a):=0$

$S:=\varnothing$

{初始化标记，a 的标记为 0，其余结点标记为∞，S 是空集}

while $z\notin S$

begin

$u:=$ 不属于 S 的 $L(u)$ 最小的一个顶点

$S:=S\cup\{u\}$

for 所有不属于 S 的顶点 v

if $L(u)+w(u,v)<L(v)$

then $L(v):=L(u)+w(u,v)$

{这样就给 S 中添加带最小标记的顶点并且更新不在 S 中的顶点的标记}

end {$L(z)=$ 从 a 到 z 的最短路的长度}

下面我们通过一个实例来说明迪克斯屈拉算法是如何工作的.

例 1 用迪克斯屈拉算法求图 2-11 所示的加权图中顶点 a 与 z 之间最短路的长度.

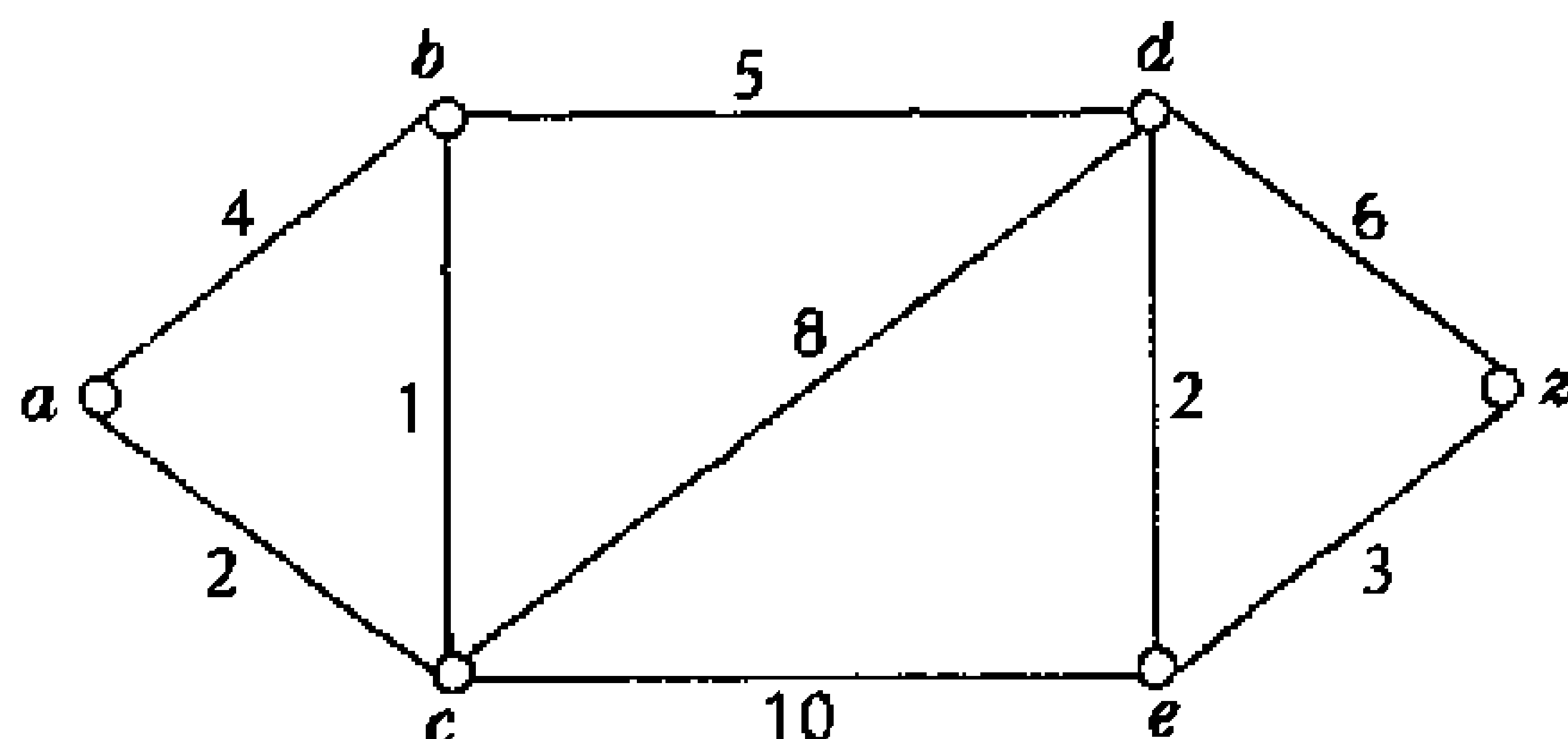


图 2-11

解 我们用表 2-3 显示用迪克斯屈拉算法求 a 和 z 之间最短路所用的步骤.

表 2-3

0 次迭代（初始化）	$L(a)=0$ $L(b)=L(c)=L(d)=L(e)=L(z)=\infty$ $S=\varnothing$
一次迭代	$u=a,s=\{a\}$
	$L(a)+w(a,b)=0+4=4<L(b)$ $L(a)+w(a,c)=0+2=2<L(c)$ $L(a)+w(a,d)=0+\infty=\infty$ $L(a)+w(a,e)=0+\infty=\infty$ $L(a)+w(a,z)=0+\infty=\infty$
	$L(b)=4,L(c)=2,L(d)=L(e)=L(z)=\infty$
二次迭代	$u=c,s=\{a,c\}$
	$L(c)+w(c,b)=2+1=3<L(b)$ $L(c)+w(c,d)=2+8=10<L(d)$ $L(c)+w(c,e)=2+10=12<L(e)$ $L(c)+w(c,z)=2+\infty=\infty$
	$L(b)=3,L(d)=10,L(e)=12,L(z)=\infty$
三次迭代	$u=b,s=\{a,c,b\}$
	$L(b)+w(b,d)=3+5=8<L(d)$ $L(b)+w(b,e)=3+\infty=\infty$ $L(b)+w(b,z)=3+\infty=\infty$
	$L(d)=8,L(e)=12,L(z)=\infty$

四次迭代	$u=d,s=\{a,c,b,d\}$
	$L(d)+w(d,e)=8+2=10<L(e)$ $L(d)+w(d,z)=8+6=14<L(z)$
	$L(e)=10,L(z)=14$
五次迭代	$u=e,s=\{a,c,b,d,e\}$ $L(e)+w(e,z)=10+3=13<L(z)$ $L(z)=13$
结　束	$u=z,s=\{a,c,b,d,e,z\}$ 从 a 到 z 的最短路的长度为 13，最短路为 (a,c,b,d,e,z).

定理 1　迪克斯屈拉算法求出连通简单无向加权图中两个顶点之间最短路的长度.

证明　当 $k=0$ 时，即没有进行任何迭代前（初始化），$L(a)=0$，$s=\{a\}$，从 a 到除 a 以外的顶点的最短路的长度是 ∞，而从 a 到 a 的最短路长度为 0.

假设在第 k 次迭代中

(1)在 s 中的顶点 v 的标记 $L(v)$ 是从 a 到这个顶点 v 的最短路的长度.

(2)不在 s 中的顶点的标记是（除了这个顶点自身之外）只经过 s 中顶点的从 a 到这个顶点的最短路的长度.

在第 $k+1$ 次迭代中

设 u 是第 k 次迭代结束时带最小标记的不在 s 中的顶点（若该顶点不唯一，可采用带最小标记的任意顶点）. 在第 $k+1$ 次迭代中，u 是添加到 s 中的顶点，则在第 $k+1$ 次迭代中，u 的标记必须是 a 到 u 的最短路的长度. 否则，k 次迭代后，有从结点 a 到某个结个 t 的路，其路的长度小于 $L_k(v)$. 下面将证明 $t\in s$. 我们用反证法，假设 $t\notin s$，设 P 是从 a 到 t 的最短路. x 为 P 上离 a 最近的不属于 s 的结点，显然 x 前面的结点都属于 s，因而存在一条从

a 到 t 的只经过 s 中顶点的长度小于 $L_k(u)$ 的路，这与 u 的选取矛盾．因此，在第 $k+1$ 次迭代结束时(1)成立(由假设，s 中 u 以外的顶点在 $k+1$ 次迭代前都已经用从 a 到该结点的最短路长度标记)．

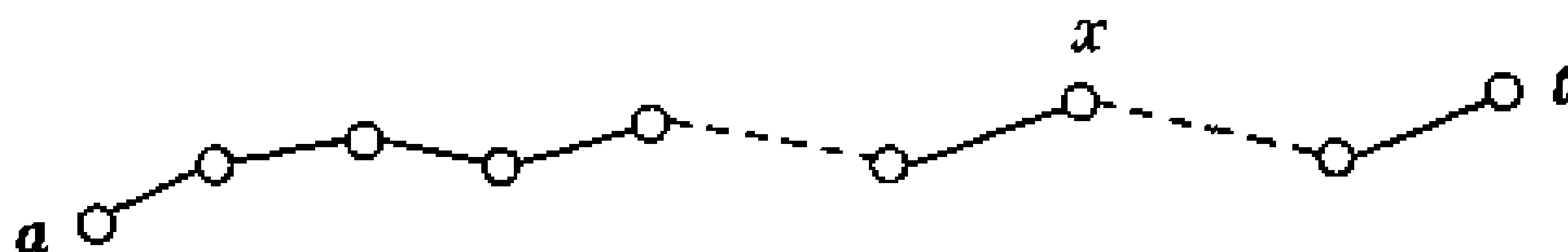


图 2-12　路 P

设 y 是第 $k+1$ 次迭代后不属于 s 的一个顶点．从 a 到 y 的只经过 s 中顶点的最短路要么包含 u，要么不包含 u．若它不包含 u，由假设，它的长度为 $L_k(y)$；若它包含 u，则它必然是这样组成的：一条从 a 到 u 的具有最短可能长度的路，其中包含 s 中不同于 u 的元素，后面接着边 $\{u, y\}$，此时，它的长度是 $L_k(u)+w(u,y)$，而 $L_{k+1}(y)=\min(L_k(y), L_k(u)+w(u,y))$．因此，第 $k+1$ 次迭代结束时(2)成立．证毕

定理 2　迪克斯屈拉算法使用 $O(n^2)$ 次运算(加法和比较)来求出 n 阶连通简单无向加权图中两个顶点之间最短路的长度．

证明　这个算法最多使用 $n-1$ 次迭代，在每次迭代中，加法运算执行的时间为 $O(n)$，比较执行的时间为 $O(n)$，因此，在每次迭代执行的时间为 $O(n)$．故该算法时间复杂度为 $O(n^2)$．证毕

求加权无向图中最短路的 Dijkstra 算法可以推广到求加权有向图中最短有向路．

已知 n 阶加权有向图 $G=\langle V, E\rangle$，其中 $V=\{1,2,\cdots,n\}$．为方便，假设 G 中不含长度非正的有向圈．若用 S_j 和 S_k 分别表示自结点 1 到结点 j 和结点 k 的最短有向路的长度，用 $w\langle k,j\rangle$ 表示有向边 $\langle k,j\rangle$ 的权(若边 $\langle k,j\rangle \notin E(G)$，则 $w\langle k,j\rangle=\infty$)，则对

一切 $j\neq k, j,k=2,3,\cdots,n$，有

$$\begin{cases}S_1=0\\ S_j\leqslant S_k+w\langle k,j\rangle\end{cases}\tag{1}$$

且 $S_j=S_k+w\langle k,j\rangle$ 当且仅当边 $\langle k,j\rangle$ 在自结点 1 到结点 j 的最短有向路上.

因为各 S_j 是 G 中的结点 1 到结点 j 的最短有向路的长度，所以这条最短有向路必有最后一条有向边 $\langle k,j\rangle$，而且该有向路上自结点 1 到 k 的一段也必是最短有向路，从而(1)式可写为

$$\begin{cases}S_1=0\\ S_j=\min\limits_{k\neq 1}(S_k+w\langle k,j\rangle), j=2,3,\cdots,n\end{cases}\tag{2}$$

这说明，图 G 中自点 1 到各点的最短有向路的长度必满足(2)式，即它们是方程(2)的解.

定理 3 设有向图 G 中不含长度非正的有向圈，并且从点 1 到其余各点都有有限长的有向路，那么式(2)有唯一有限解 $\{S_j\}$.

证明 设 $\{S_1,S_2,\cdots,S_n\}$ 为式(2)的任一组有限解. 对任意 S_j，我们可以在 G 中找出结点 1 到结点 j 的有向路.

$$P=1i_1i_2\cdots i_{k-1}i_kj$$

使得

$$\begin{aligned}&S_{i_1}=S_1+w\langle 1,i_1\rangle\\ &S_{i_2}=S_{i_1}+w\langle i_1,i_2\rangle\\ &\cdots\\ &S_{i_k}=S_{i_{k-1}}+w\langle i_{k-1},i_k\rangle\\ &S_j=S_{i_k}+w\langle i_k,j\rangle\end{aligned}$$

设 $S_1,S_2,\cdots,S_n$ 是自结点 1 到各结点的最短有向路的长度，因此，它是式(2)的一组解. 若式(2)的解不唯一，设 $S'_1,S'_2,\cdots,S'_n$ 是不同于 $S_1,S_2,\cdots,S_n$ 的另一组解，那么必存在某个 j，使 S_j

$\neq S'_j$. S_j 是自结点 1 到结点 j 的最短有向路的长度，而 S'_j 是自结点 1 到结点 j 的某一有向路的长度，因此有 $S'_j > S_j$. 在 $S'_j > S_j$ 的所有 j 中，总可以选取某个 j，使边 $\langle k,j\rangle$ 是自结点 1 到结点 j 最短有向路上的边，且 $S'_k = S_k$（因为 $S'_1 = S_1$，所以这样的有向边一定存在），从而有

$$S'_j > S_j = S_k + w\langle k,j\rangle = S'_k + w\langle k,j\rangle$$

这与 $S'_1, S'_2, \cdots, S'_n$ 为式(2)的解相矛盾. 证毕

定理 3 说明式(2)的解是自结点 1 到其余各结点的最短有向路的长度. 但是，直接求解方程式(2)是很困难的. 目前几乎所有求最短有向路的算法，都是围绕着怎样解这个方程的问题. 只有对于某些特殊的有向图，方程式(2)可以简化，从而变得较易求解.

定理 4 设 S_j 是加权有向图 G 中自结点 1 到结点 j 的最短有向路的长度，并且对所有的 $j = 1,2,3,\cdots,n$, S_j 为有限值. 若图 G 中除结点 1 外的其余各点能重新编写成如下的序号 $2,3,\cdots,n$，使得对所有 $i<j$，成立

$$S_i \leqslant S_j \text{ 且 } w\langle j,i\rangle \geqslant 0$$

或者 $S_i > S_j$ 且 $w\langle j,i\rangle = \infty$，即 $\langle j,i\rangle \notin E(G)$

则方程式(2)可简化为

$$\begin{cases} S_1 = 0 \\ S_j = \min\limits_{k<j}(S_k + w\langle k,j\rangle), j = 2,3,\cdots,n \end{cases} \tag{3}$$

显然，式(3)比式(2)更容易求解，因为 $S_1 = 0$ 是已知的，而 S_2 只依赖于 S_1, S_3 只依赖于 S_2 和 S_1, $\cdots$, S_k 只依赖于 $S_{k-1}, S_{k-2}, \cdots, S_2$ 和 S_1. 当然，找出有向图 G 中各结点满足定理 4 的顺序是问题的关键.

一个边权为正值的有向图显然具有如下性质：

定理 5 设 $G = \langle V,E\rangle$ 是一个边权为正值的有向图，其中 $V = \{1,2,3\cdots,n\}$. 则在 G 中，任意一条最短有向路的长都大于它

的真子有向路的长.

根据定理 5 知,G 中自结点 1 到其他各结点最短有向路的长可按大小排列如下:

$$0=S_1\leqslant S_2\leqslant\cdots\leqslant S_n$$

于是,它满足定理 4 的条件,因此可以用方程式(3)来求解.下面给出在边权为正值的有向图中求最短向路的 Dijkstra 算法.

(1)置 $P=\{1\}, T=\{2,3,\cdots,n\}$ 且 $S_1=0, S_j=w\langle 1,j\rangle, j=2,3,\cdots,n$.

(2)在 T 中寻找一点 k,使得 $S_k=\min\limits_{j\in T}\{S_j\}$,置 $P=P\cup\{k\}$, $T=T-\{k\}$.若 $T=\varnothing$,终止.否则,转向(3).

(3)对 T 中每一点 j,置 $S_j=\min\{S_j, S_k+w\langle k,j\rangle\}$,然后转向(2).

显然该算法经过 $n-1$ 次循环必结束.

Dijkstra 算法只求出图中一个特定顶点到其它各顶点的最短路.但在许多实际问题中,需求出任意两点之间的最短路,如全国各城市之间最短的航线,选址问题等.当然,要求出一个图的任意两点间的最短路,只需将图中每一个顶点依次视为始点,然后用 Dijkstra 算法就可以,但这要用到大量的计算.下面我们先简要介绍一个比较好的算法——Floyd(1962)算法.

为了讨论方便,先定义矩阵的一种运算.

定义 1 已知矩阵 $A=(a_{ij})_{m\times l}, B=(b_{jk})_{l\times n}$,规定 $C=A*B=(C_{ij})_{m\times n}$,其中

$$C_{ij}=\min(a_{i1}+b_{1j}, a_{i2}+b_{2j}, \cdots, a_{il}+b_{lj})$$

定义 2 已知矩阵 $A=(a_{ij})_{m\times n}, B=(b_{ij})_{m\times n}$,规定 $D=A\circledast B=(d_{ij})_{m\times n}$,其中

$$d_{ij}=\min(a_{ij}, b_{ij})$$

可以利用上面定义的运算求任意两点间的最短距离.

已知 n 阶加权简单图G,设 $D=(d_{ij})_{n\times n}$是图G 的边权矩阵,即 $d_{ij}=w(i,j)$(若 G 是有向图,则 $d_{ij}=w\langle i,j\rangle$),若结点 i 到结点 j 无边相连时,则取 $d_{ij}=\infty$.

然后依次计算出矩阵 $D^{[2]},D^{[3]},\cdots,D^{[n]}$及 S.

其中

$$D^{[2]}=D*D=(d_{ij}^{[2]})_{n\times n}$$

$$D^{[3]}=D^{[2]}*D=(d_{ij}^{[3]})_{n\times n}$$

……

$$D^{[n]}=D^{[n-1]}*D=(d_{ij}^{[3]})_{n\times n}$$

$$S=D\circledast D^{[2]}\circledast D^{[3]}\circledast\cdots\circledast D^{[n]}=(S_{ij})_{n\times n}$$

由定义可知,$d_{ij}^{[k]}$表示从结点 i 到 j 经 k 边的路(在有向图中即为有向路)中的长度最短者,而 S_{ij}为结点 i 到 j 的所有路(若是有向图,即为有向路)中的长度最短者.

不难看出,Floyd 算法的时间复杂性是 $O(n^4)$.下面看介绍求任意两点间最短路的 Warshall 算法.

已知 n 阶加权简单图G,设 $D=(d_{ij})_{n\times n}$是图G 的边权矩阵.

(1)输入 D;

(2)$k:=1$;

(3)$i:=1$;

(4)$d_{ij}:=\min(d_{ij},d_{ik}+d_{kj}),j=1,2,\cdots,n$;

(5)$i:=i+1$,若 $i\leqslant n$,转(4);

(6)$k:=k+1$,若 $k\leqslant n$,转(3);否则停止.

该算法是对 i,j,k 进行循环,故它的复杂性是 $O(n^3)$,即对矩阵 D 进行 k 次修改.

例 2 求图 2-13 中任意两点间最短有向路的长度.

解 图 G 的边权矩阵D 为(结点顺序为 v_1,v_2,v_3,v_4,v_5,v_6):

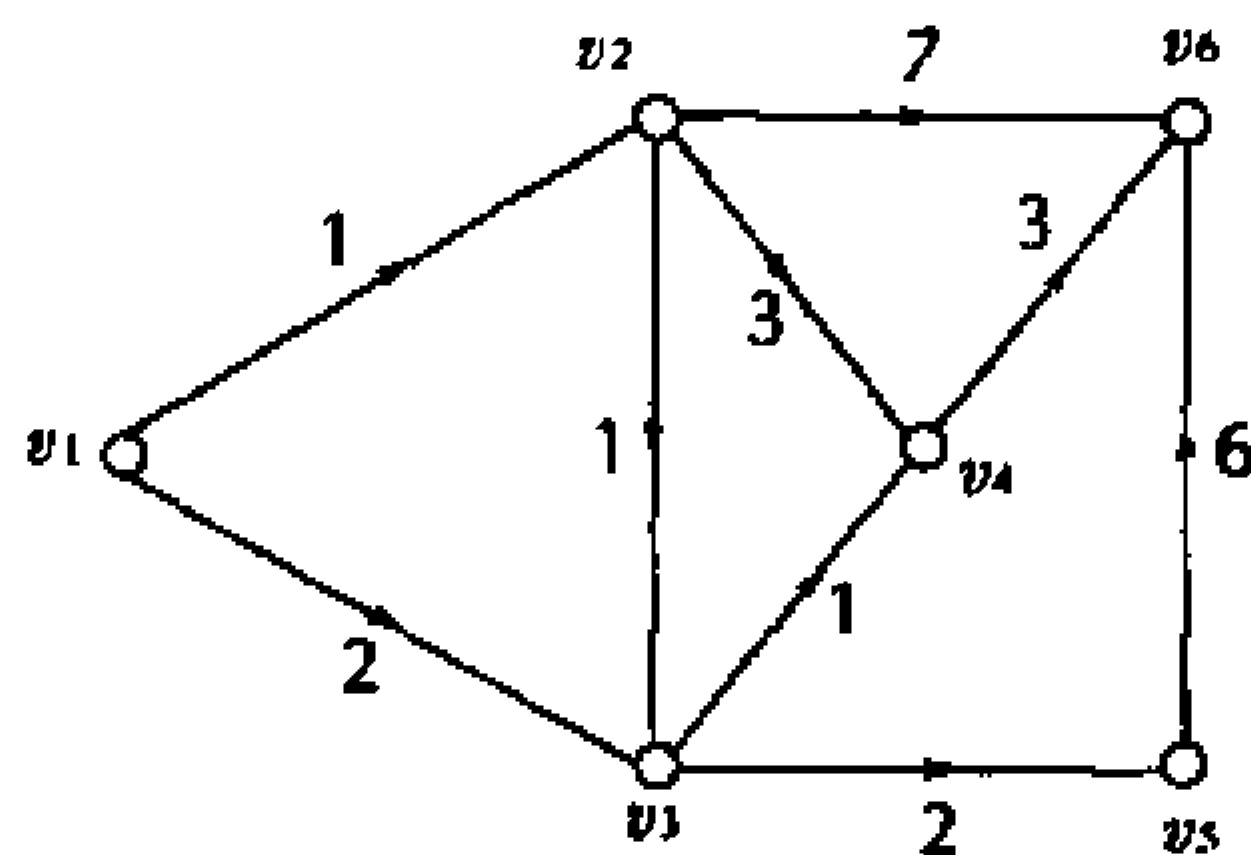


图 2-13

$$
D=\begin{bmatrix}
\infty & 1 & 2 & \infty & \infty & \infty \\
\infty & \infty & 1 & 3 & \infty & 7 \\
\infty & \infty & \infty & 1 & 2 & \infty \\
\infty & \infty & \infty & \infty & \infty & 3 \\
\infty & \infty & \infty & \infty & \infty & 6 \\
\infty & \infty & \infty & \infty & \infty & \infty
\end{bmatrix}
$$

$$
D^{[2]}=\begin{bmatrix}
\infty & 1 & 2 & \infty & \infty & \infty \\
\infty & \infty & 1 & 3 & \infty & 7 \\
\infty & \infty & \infty & 1 & 2 & \infty \\
\infty & \infty & \infty & \infty & \infty & 3 \\
\infty & \infty & \infty & \infty & \infty & 6 \\
\infty & \infty & \infty & \infty & \infty & \infty
\end{bmatrix} *
\begin{bmatrix}
\infty & 1 & 2 & \infty & \infty & \infty \\
\infty & \infty & 1 & 3 & \infty & 7 \\
\infty & \infty & \infty & 1 & 2 & \infty \\
\infty & \infty & \infty & \infty & \infty & 3 \\
\infty & \infty & \infty & \infty & \infty & 6 \\
\infty & \infty & \infty & \infty & \infty & \infty
\end{bmatrix}
$$

$$
=\begin{bmatrix} \infty & \infty & 2 & 3 & 4 & 8 \\ \infty & \infty & \infty & 2 & 3 & 6 \\ \infty & \infty & \infty & \infty & \infty & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}
$$

同理可得

$$
D^{[3]}=\begin{bmatrix} \infty & \infty & \infty & 3 & 4 & 6 \\ \infty & \infty & \infty & \infty & \infty & 5 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}
$$

$$
D^{[4]}=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}
$$

$D^{[5]}=(d_{ij}^{[5]})_{n\times n}$，其中 $d_{ij}^{[5]}=\infty$.

$D^{[6]}=(d_{ij}^{[6]})_{n\times n}$，其中 $d_{ij}^{[6]}=\infty$.

$\therefore S=D\circledast D^{[2]}\circledast D^{[3]}\circledast D^{[4]}=(S_{ij})_{n\times n}$

$$
=\begin{bmatrix} \infty & 1 & 2 & 3 & 4 & 6 \\ \infty & \infty & 1 & 2 & 3 & 5 \\ \infty & \infty & \infty & 1 & 2 & 4 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}
$$

由 $S_{16}=6$ 知，从结点 v_1 到 v_6 的最短有向路的长度为 6；由 $S_{35}=2$ 知，从结点 v_3 到 v_5 的最短有向路的长度为 2；由 $S_{45}=\infty$ 知，从结点 v_4 到 v_5 没有有向路. 等等.

已知 G 的边权矩阵 D，运用 Warshall 算法得

$k=1$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & \infty & \infty & \infty \\ \infty & \infty & 1 & 3 & \infty & 7 \\ \infty & \infty & \infty & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$k=2$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & 4^* & \infty & 8^* \\ \infty & \infty & 1 & 3 & \infty & 7 \\ \infty & \infty & \infty & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$k=3$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & 3^* & 4^* & 8 \\ \infty & \infty & 1 & 2^* & 3^* & 7 \\ \infty & \infty & \infty & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$k=4$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & 3 & 4 & 6^* \\ \infty & \infty & 1 & 2 & 3 & 5^* \\ \infty & \infty & \infty & 1 & 2 & 4^* \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$k=5$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & 3 & 4 & 6 \\ \infty & \infty & 1 & 2 & 3 & 5 \\ \infty & \infty & \infty & 1 & 2 & 4 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$k=6$ 时

$$D=\begin{bmatrix} \infty & 1 & 2 & 3 & 4 & 6 \\ \infty & \infty & 1 & 2 & 3 & 5 \\ \infty & \infty & \infty & 1 & 2 & 4 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

第六节　旅行推销员问题和中国投递员问题

旅行推销员问题(travelling salesman problem)是求加权完全无向图中的访问每个顶点恰好一次且返回出发点的总权数最小的闭路,又称之为**最优哈密顿回路**.如果我们将加权图中的结点看作城市,加权边看作距离,旅行推销员问题就成为找出一条最短路线,使得旅行推销员从某个城市出发,遍历每个城市一次,最后再

回到出发的城市.

旅行推销员问题等价于求完全加权图中总权数最小的哈密顿回路.若选定出发点,对 n 个城市进行排列,因第二个顶点有 $n-1$ 种选择,第三个顶点有 $n-2$ 种选择,依次类推,共有 $(n-1)!$ 条哈密顿回路.考虑经相反顺序来经过一条哈密顿回路,因而需要检查 $\frac{(n-1)!}{2}$ 条哈密顿回路,从中找出权和最小的一条.我们知道 $\frac{(n-1)!}{2}$ 增长得极快.比如有 25 个顶点,需考虑 $\frac{24!}{2}$(约为 3.1×10^{23})条不同的哈密顿回路.假定检查每条哈密顿回路只花费 1 纳秒(10^{-9}秒),那么就需大约 1 千万年才能求出该图中长度最短的一条哈密顿回路.

因为旅行推销员问题同时具有实践和理论的重要性,所以已经投入了巨大的努力来设计解决它的有效算法.目前还没有找到一个有效算法,现已证明它是属于 *NPC* 问题.

当有许多需要访问的顶点时,解决旅行推销员问题的实际方法是使用**近似算法(approximation algorithm)**.

下面介绍简便的"最邻近方法"给出旅行推销员问题的近似解.

最邻近方法的步骤如下:

(1)由任意选择的结点开始,指出与该点最靠近(即权最小)的点,形成有一条边的初始路.

(2)设 x 表示最新加到这条路上的结点,从不在路上的所有结点中选一个与 x 最靠近的结点,把连接 x 与这一结点的边加到这条路上.重复这一步,直到图中所有结点包含在路上.

(3)将连接起点与最后加入的结点之间的边加到这条路上,就得到一个哈密顿回路,即得问题的近似解.

例 1 用"最邻近方法"找出图 2-14 所示加权完全图中具有

充分小权的哈密顿回路.

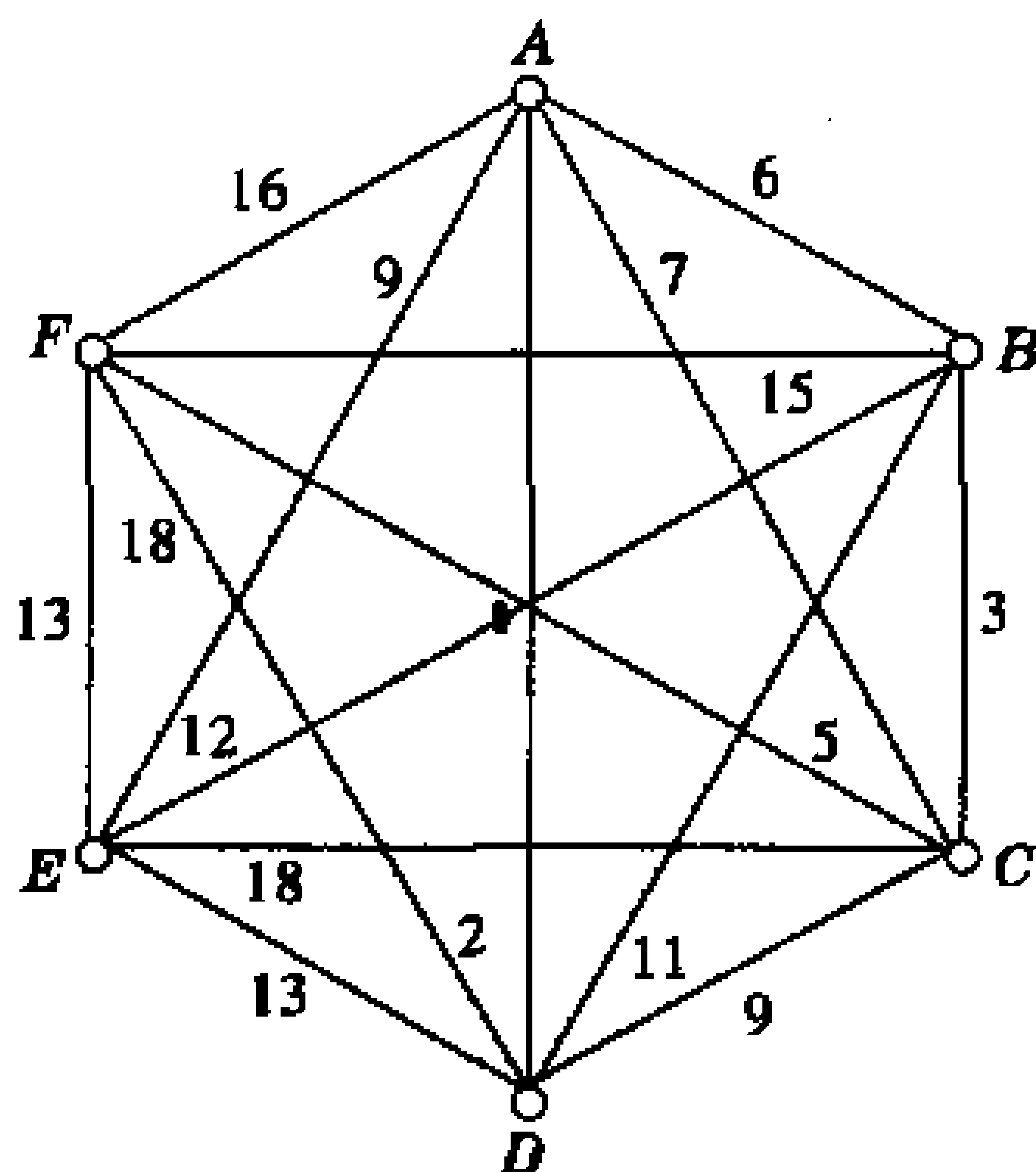


图 2-14

解 *ADCBEFA* 权和 55

BCADEFB 权和 53

CBADEFC 权和 42

DABCFED 权和 42

EADCBFE 权和 51

FCBADEF 权和 42

由上例可知,所选取的哈密顿回路不同,其近似解也不同.而"最邻近插入方法"对上述方法可以进行改进,从而产生一个较好的结果.该方法在每次迭代中都构成一个闭的旅行路线.它是由多个阶段而形成的一个个旅程,逐步建立起来的,每一次比上一次多一个顶点,即是说,下一个旅程总比上一个旅程多一个顶点.求解时,在已建立旅程以外的顶点中,寻找最邻近于旅程中某个顶点的顶点,然后将其插入该旅程中,并使增加的距离尽可能小,当全部顶点收入这个旅程后,就找到了我们所求的最短哈密顿回路的近

似解.

例 2 用"最邻近插入方法"找出图 2-14 所示加权完全图中具有充分小权的哈密顿回路.

解 开始于顶点 A,组成闭旅程 AA.在下一阶段,最邻近 A 的顶点为D,建立闭旅程 ADA.顶点 B 最邻近顶点A,建立闭旅程 ADBA.

接下来,由于 C 最邻近 B,将 C 插入,分别得到三个闭旅程 ACDBA、ADCBA、ADBCA,其长度依次为 33,20,23,选取长度最短的旅程 ADCBA.

继而,距旅程 ADCBA 中顶点最邻近顶点为 F,将 F 插入,分别得到四个闭旅程 AFDCBA、ADFCBA、ADCFBA、ADCBFA,其长度依次为 52,34,37,45,选取长度最短的旅程 ADFCBA.

最后,把顶点 E 插入旅程 ADFCBA 中,得到五个闭旅程 AEDFCBA, ADEFCBA, ADFECBA, ADFCEBA, ADFCBEA,其长度依次为 54,42,60,61,49.显然,长度最短的旅程 $ADEFCBA$ 即为我们要求的最短哈密顿回路的近似解.

中国投递员问题(Chinese Postman Problem)是由我国管梅谷教授首先提出并加以研究的.投递员在邮局分拣好需要投递的邮件后,到他管辖的区域内的每一条街道投递,最后返回邮局.对于这个问题,投递员希望找出一条行走距离最短的路线.

我们把投递员所管辖的投递区域看作一个连通的加权无向图 G,其中 G 的顶点看作街道的交叉口和端点,街道看作边,权看作街道的长度.解决中国投递员问题,就是在连通加权无向图中,寻找一条经过每边至少一次且权和最小的闭链,即 G 的**最优环游**.

如果对应的图 G 是欧拉图,那么从对应于邮局的顶点出发的任何一条欧拉回路都是符合上述要求的投递员的最优投递路线;如果图 G 只有两个奇度数顶点 x 和 y,则存在一条以 x 和 y 为端点的欧拉路,因此,所要求的最优投递路线是由这条欧拉路 + 边

$\{x,y\}$.

设连通图 G 不是欧拉图.由于图 G 有偶数个奇度数顶点,对于任两个奇度数顶点 x 和 y,在 G 中必有一条路连结它们.将这条路上的每条边改为二重边得到新图 H_1,则 x 和 y 就变为 H_1 的偶度数顶点,在这条路上的其它顶点的度数均增加 2,即奇偶性不变,于是 H_1 的奇顶点个数比 G 的奇顶点个数少 2.对 H_1 重复上述过程得 H_2,再对 H_2 重复上述过程得 $H_3,\cdots$,经若干次后,可将 G 中所有奇度数顶点变成偶度数顶点,从而得到多重欧拉图 G'(在 G' 中,若某两点 u 和 v 之间连接的边数多于 2,则可去掉其中的偶数条多重边,最后剩下连接 u 与 v 的 1 或 2 条边,这样得到的图仍是 Euler 图).这个欧拉图 G' 的一条欧拉回路就相应于中国投递员问题的一个可行解,且欧拉回路的长度等于 G 的所有边的长度加上每次连奇度数顶点的路的长度.但怎样才能使这样的闭链的长度最短呢?亦即如何求 G 的最优环游呢?

定理 1 设 P 是加权连通图 G 中一条包含 G 的所有边至少一次的闭链,则 P 最优(即具有最小长度)的充要条件是:

(1) P 中没有二重以上的边;

(2) 在 G 的每条圈 C 中,重复边集 E 的长度之和不超过这个圈长度的一半,即 $w(E)\leqslant\frac{1}{2}w(C)$.

证明 (⇒)首先可以肯定 P 中不存在重数大于 2 的边.假设 G 有一条长度大于 2 的圈 C_1,且 C_1 中重复边的权和大于 C_1 的权和的一半,则将 C_1 中原来无重复边的边各添加一条重复边,而将 C_1 上的各重复边分别删去一条边,这样重复边的长度之和减少,而 Euler 图的性质不变.这与 P 的最优性矛盾.

(⇐)只要证明满足定理条件(1),(2)的所有闭链的长度相等.因为这些闭链要包含 G 的所有边,所以只要证明重复边的长度之和相等即可.

设 P_1 和 P_2 是两条满足定理条件(1),(2)的闭链,由于 P_1 和 P_2 可能有相同的重复边,只要比较 P_1 和 P_2 中不相同的重复边即可.记 P_1 和 P_2 中重复边的集合分别为 E_1 和 E_2,只要比较 $w(E_1-E_2)$和 $w(E_2-E_1)$.

记边集 $F=(E_1-E_2)\cup(E_2-E_1)$的导出子图为 G'.

设任意 $v\in V(G)$.若 $\deg_G(v)$为奇数,则 E_1 和 E_2 中均有奇数条边与 v 关联(这是因为由 G 加入重复边后得到的图是 Euler 图,即每个顶点的度均为偶数).若 $\deg_G(v)$为偶数,则 E_1 和 E_2 中与 v 关联的边数均为偶数.因此,在任何情况下,E_1 和 E_2 中与 v 关联的边数的奇偶性相同.设 E_1 和 E_2 分别有 y_1 和 y_2 条边与 v 关联,其中有 y_0 条边同时属于 E_1 和 E_2,则 F 中和 v 关联的边数为$(y_1-y_0)+(y_2-y_0)=y_1+y_2-2y_0$.由于 y_1 和 y_2 奇偶性相同,所以 F 中与 v 关联的边数为偶数,即 G'的每个连通分支是 Euler 图.因而 G 可分解成若干个圈,在每一个圈上,由条件(2),属于 E_1 的边的长度之和与属于 E_2 的边的长度之和都不超过圈长的一半.又因为圈上的边或者属于 E_1 或者属于 E_2,因此每个圈上 E_1-E_2 和 E_2-E_1 的边的长度之和相同,故可知在 G'中,$w(E_1-E_2)=w(E_2-E_1)$.证毕

根据上面的讨论及定理 1,我们可以设计出求非 Euler 赋权连通图 G 的最优环游的算法.此算法称为最优球游的**奇偶点图上作业法**.

(1)把 G 中所有奇度数顶点配成对,将每对奇度数顶点之间的一条路上的每边改为二重边,得到一个新图 G_1,新图 G_1 中没有奇度数顶点,即 G_1 为多重 Euler 图.

(2)若 G_1 中某一对顶点之间有多于 2 条边连接,则去掉其中的偶数条边,留下 1 条或 2 条边连接这两个顶点.直到每一对相邻

顶点至多由 2 条边连接. 得到图 G_2.

(3)检查 G_2 的每一个圈 C,若某一个圈 C 上重复边的权和超过此圈权和的一半,则将 C 按定理 1 必要性的证明过程进行调整. 重复这一过程,直到对 G_2 的所有圈,其重复边的权和不超此圈权和的一半,得到图 G_3.

(4)用 Fleury 算法求 G_3 的 Euler 回路.

例 3 求图 2-15 所示图 G 的最优环游.

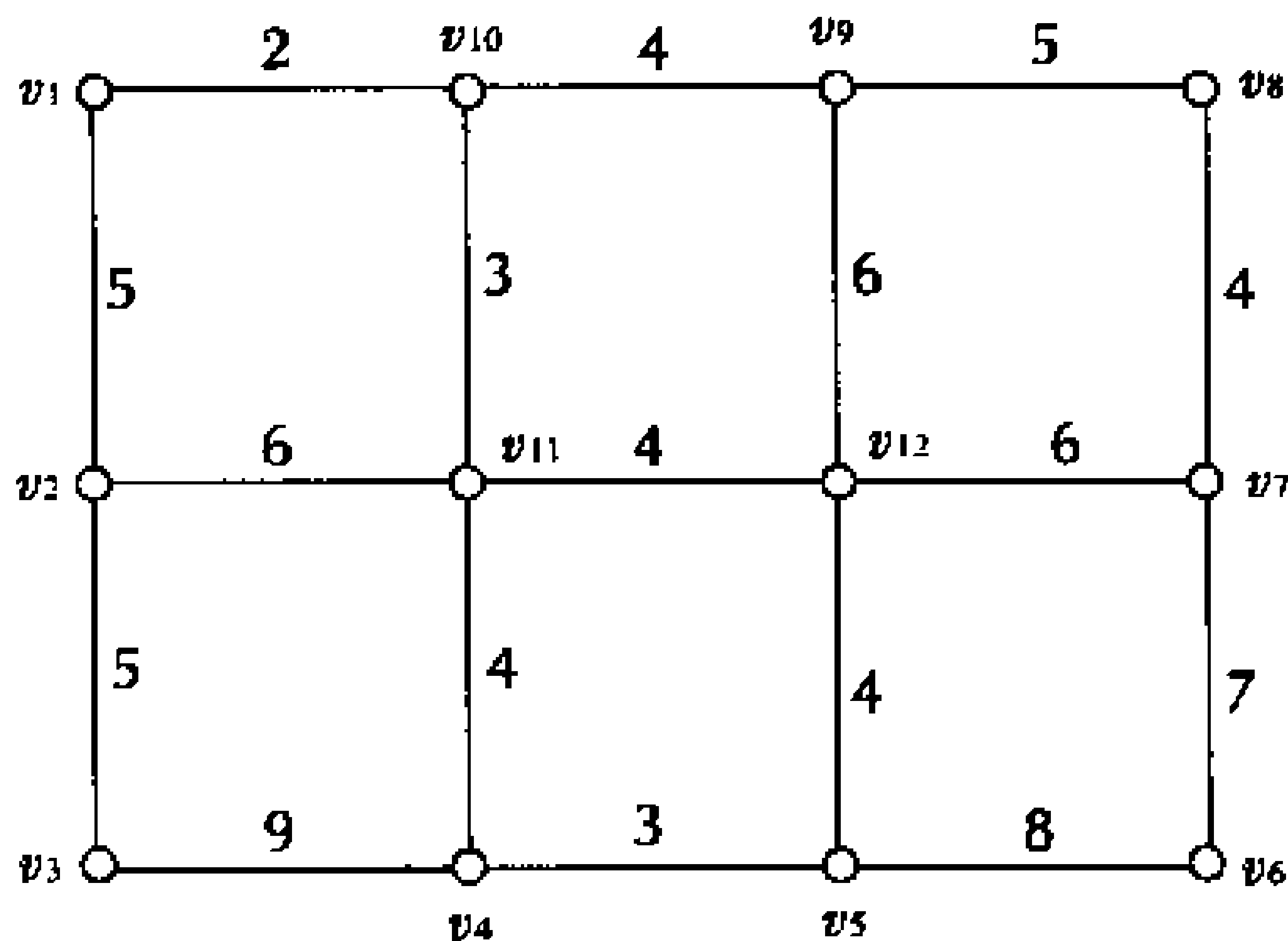


图 2-15

解 图 G 中有 6 个奇度数顶点 $v_2, v_4, v_5, v_7, v_9, v_{10}$,把它们配成三对: v_2 与 v_5, v_4 与 v_7, v_9 与 v_{10}. 在图 G 中,取一条连接 v_2 与 v_5 的路 $v_2v_3v_4v_5$,把边 $\{v_2, v_3\}$, $\{v_3, v_4\}$, $\{v_4, v_5\}$ 作为重复边加入图中;再取 v_4 与 v_7 之间一条路 $v_4v_5v_6v_7$,把边 $\{v_4, v_5\}$, $\{v_5, v_6\}$, $\{v_6, v_7\}$ 作为重复边加入图中,在 v_9 和 v_{10} 之间加一条重复边 $\{v_9, v_{10}\}$,如图 2-16 所示,这个图没有奇度数点,是一个 Euler 图.

在图 2-16 中,顶点 v_4 与 v_5 之间有三条重边,去掉其中两条,得图 2-17 所示的图,该图仍是一个 Euler 图.

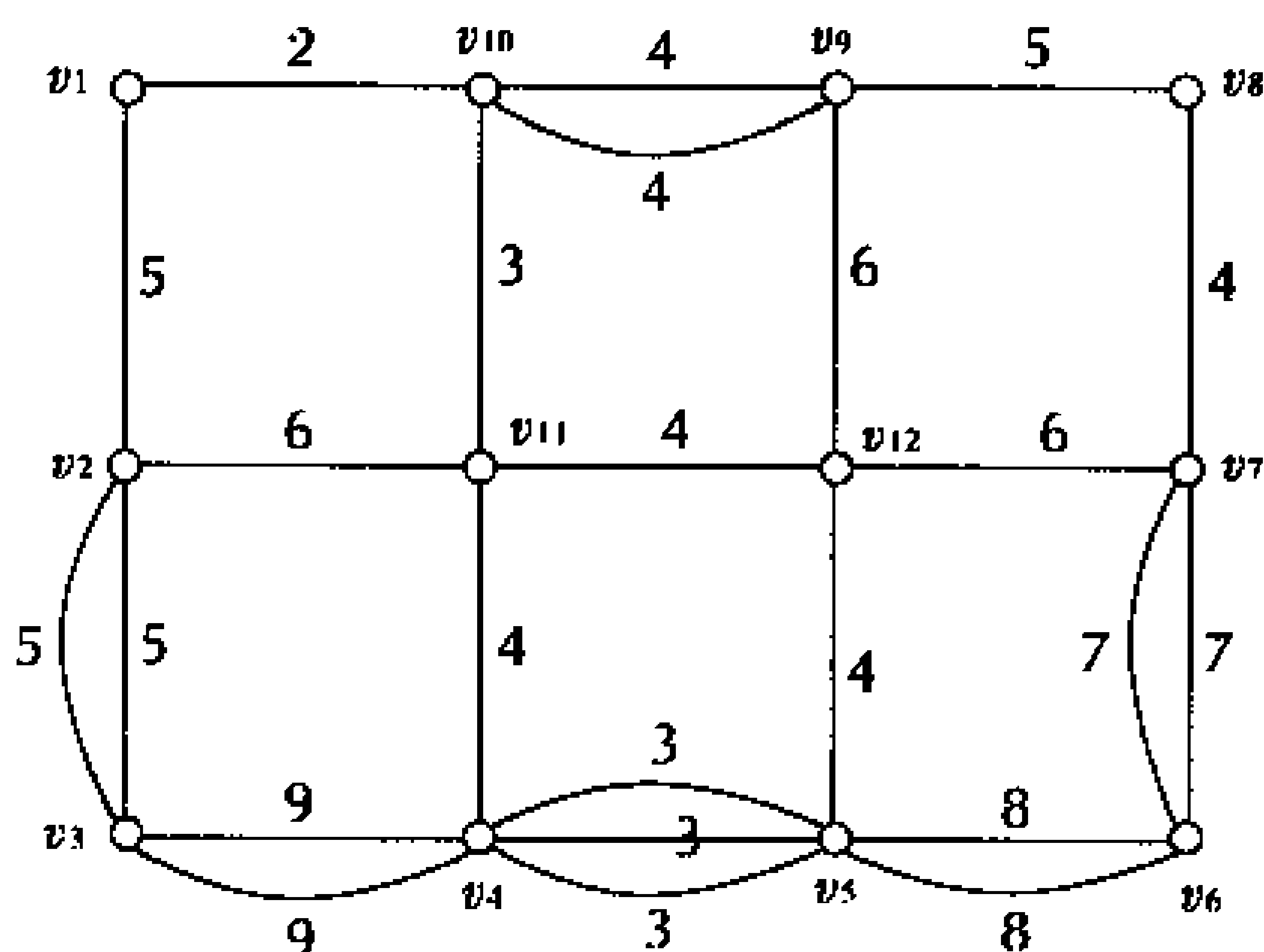


图 2-16

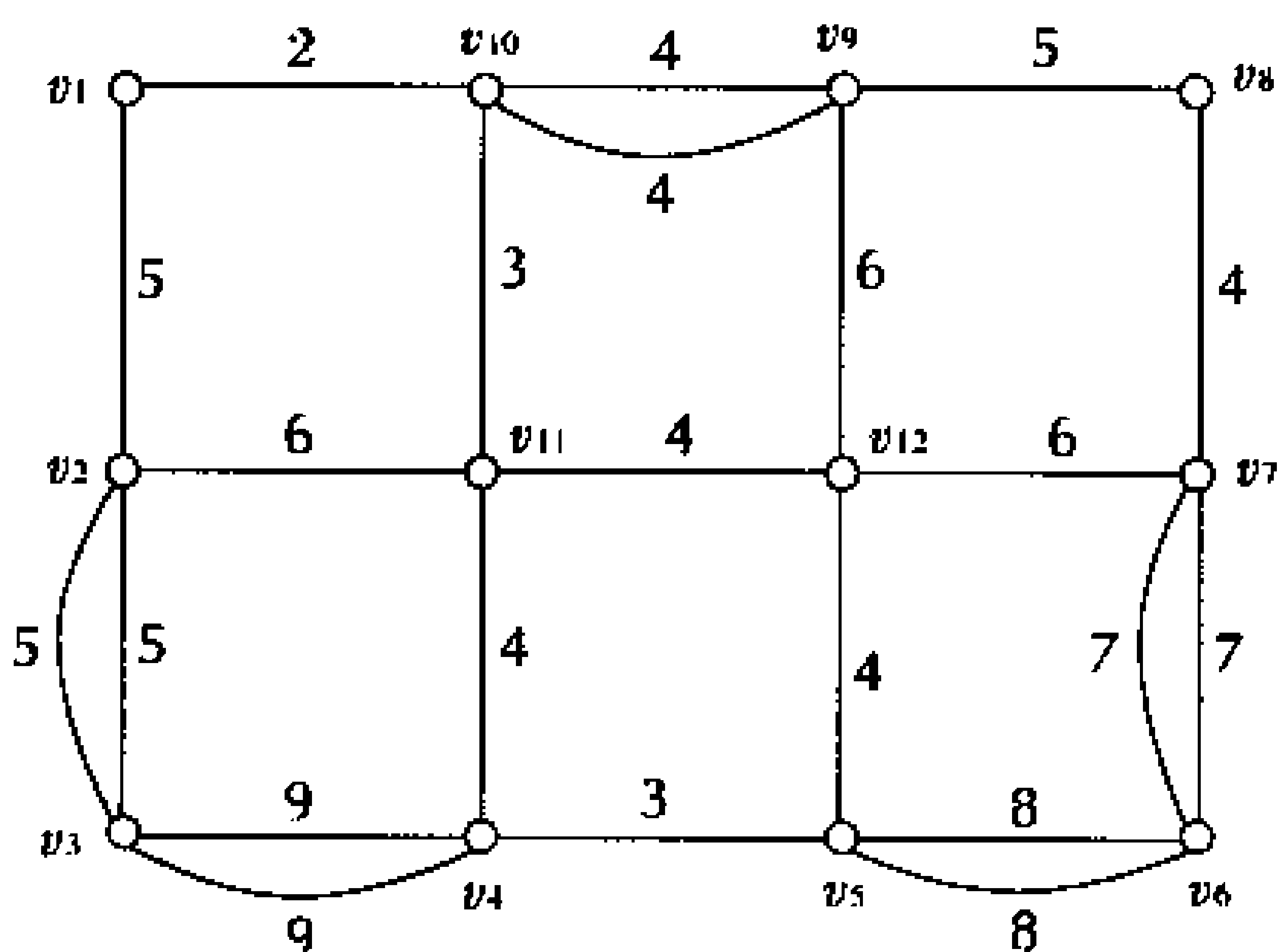


图 2-17

如图 2-17 中,圈 $v_2v_3v_4v_{11}v_2$ 的总权为 24,而圈上重复边的权和为 14,大于该圈总权的一半,于是去掉边 $\{v_2,v_3\}$ 和 $\{v_3,v_4\}$ 上的重复边,而在边 $\{v_2,v_{11}\}$ 和 $\{v_4,v_{11}\}$ 上加入重复边,此时重复边的权和为 10,小于该圈总权的一半. 同理,圈 $v_5v_6v_7v_{12}v_5$ 的总权为 25,而重复边权和为 15,于是去掉边 $\{v_5,v_6\}$ 和 $\{v_6,v_7\}$ 上的

重复边，在边$\{v_5, v_{12}\}$和$\{v_7, v_{12}\}$上加重复边，如图 2-18 所示.

图 2-18

图 2-18 中，圈 $v_4 v_5 v_{12} v_{11} v_4$ 的总权为 15，而重复边的权和为 8，从而调整为图 2-19 所示.

图 2-19

图 2-19 中，圈 $v_1 v_2 v_{11} v_{12} v_7 v_8 v_9 v_{10} v_1$ 的总权为 36，而重复边的总权为 20，继续调整为图 2-20 所示.

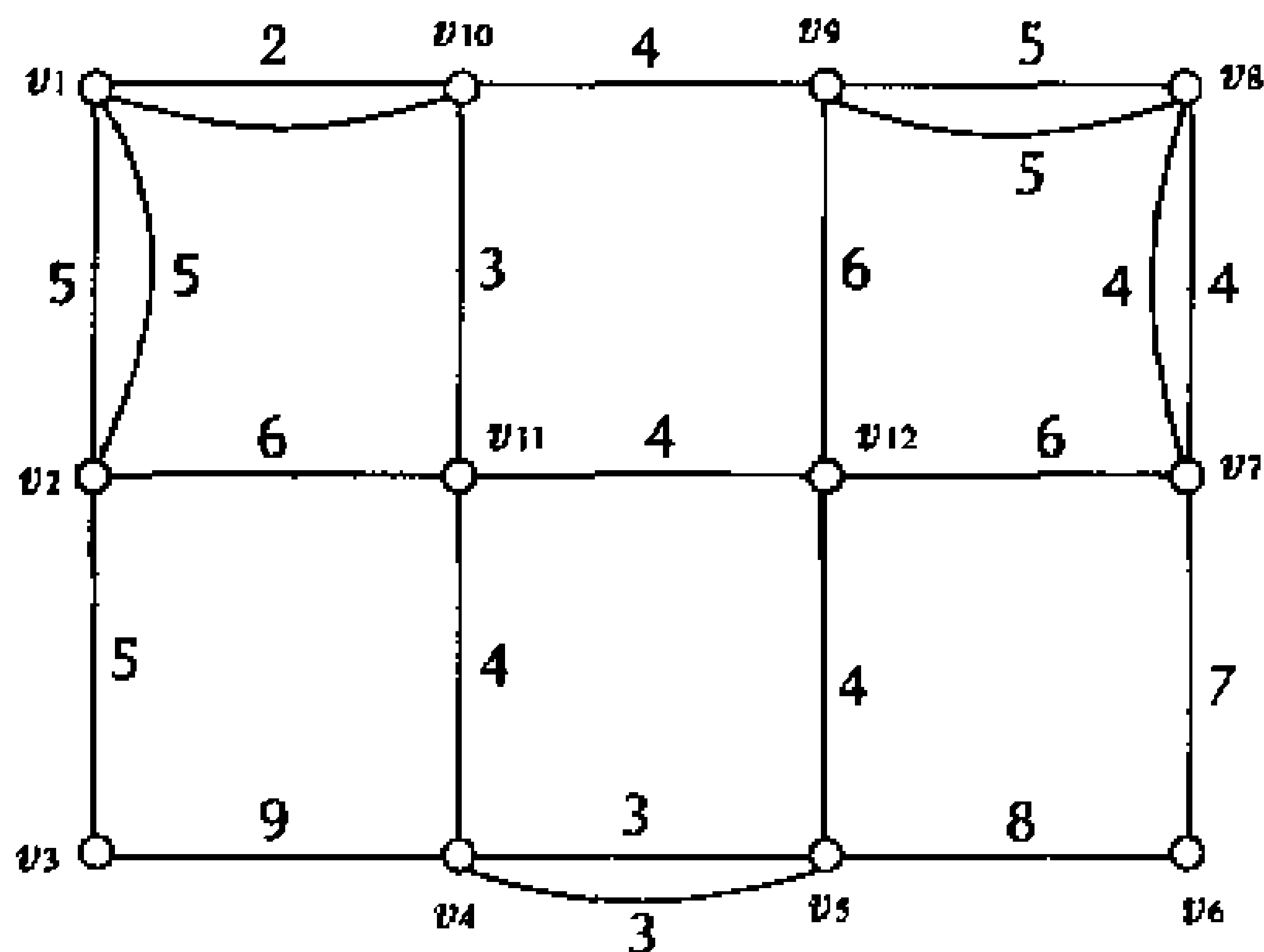


图 2-20

检查图 2-20，可知定理 1 的(1)和(2)均满足，故为最优方案。接着用 Fleury 算法求出图 2-20 所示图的 Euler 回路，即为图 G 的最优环游。

由上例可知，对于比较大的图，要考察每个圈上重复边权和不大于该圈总权和之一半，确定每个圈的时间复杂度太大。1973 年，Edmonds 和 Johnson 给出了下面的有效算法。

Edmonds－Johnson 算法：

设 G 是连通加权图

(1)求 $V_0=\{v \mid v \in V(G), d(v)$为奇数$\}$；

(2)对每对顶点 $u, v \in V_0$，求 $d(u,v)$($d(u,v)$表示 u 与 v 的距离，即 u 与 v 之间最短路的长度，可由 Dijkstra 算法求得)；

(3)构作完全图 $K_{|V_0|}$，且加权，使 $K_{|V_0|}$ 以 V_0 为顶点集，且 $u, v \in V_0$ 时，边$\{u,v\}$的权为 $d(u,v)$；

(4)求 $K_{|V_0|}$ 中权最小的完备匹配 M；

(5)在 G 中求以 M 中每条边的两端点为端点的最短路；

(6)把(5)中求得的每条最短路上每条边都添加一条等权的

"倍边";

(7)在(6)中得到的图 G' 上用 Fleury 算法求 Euler 回路，即为 G 的最优环游.

Edmonds－Johnson 算法的时间复杂度为 $O|E(G)|$.

习 题 二

1. 某企业使用一台设备，每年年初，企业要考虑是购买新设备，还是继续使用旧设备. 若购置新设备，就要支付购买费，若使用旧的，就要支付一笔维修费. 具体需要多少，根据该设备使用的年数决定. 表 2－4 给出了企业对该种设备的维修费用，表 2－5 给出了在不同年代购买该种设备所需的费用，试制定该企业五年内的设备更新计划，使得支付的费用最小.

表 2－4　　　　(单位:万元)

使用年数	0－1	1－2	2－3	3－4	4－5
维修费用	5	6	8	11	18

表 2－5　　　　(单位:万元)

第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
11	11	12	12	13

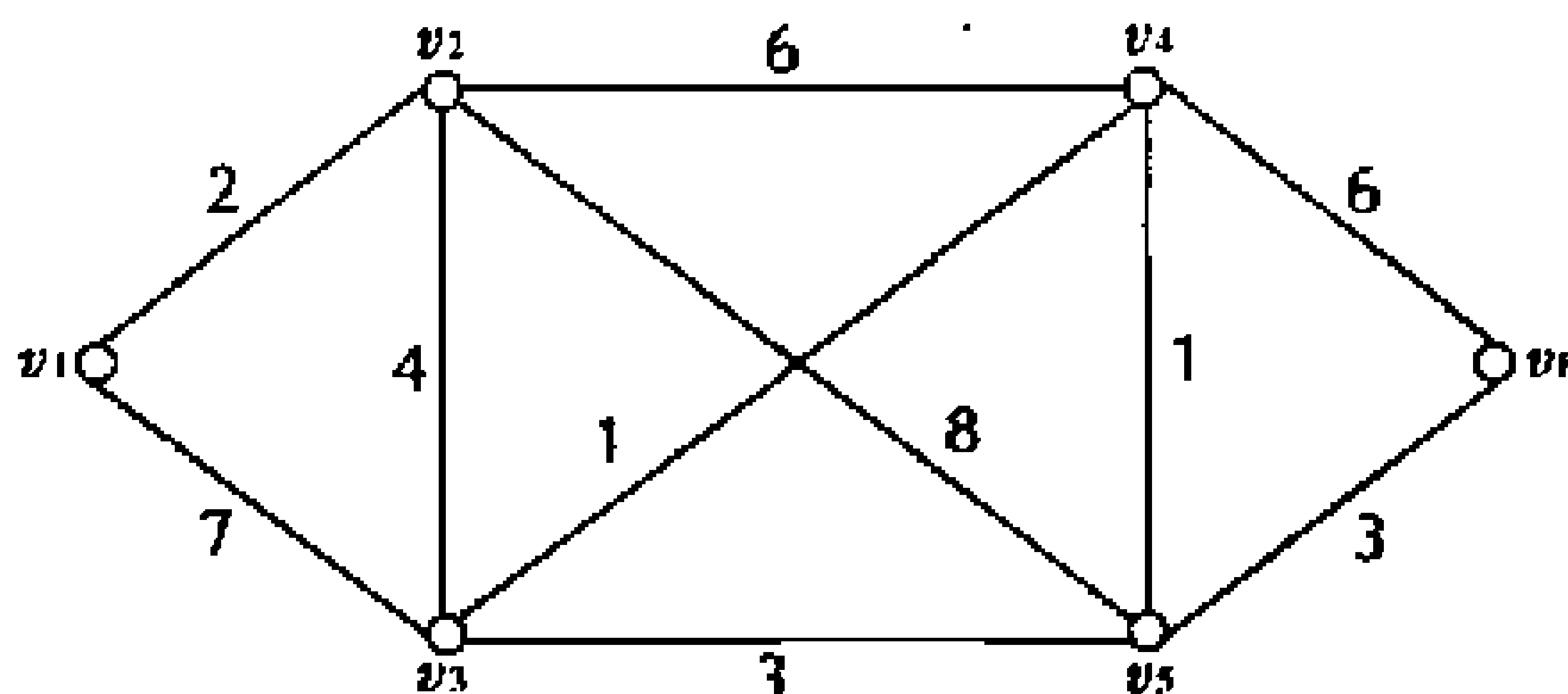


图 2-21

2. 已知有 6 个村庄，各村的小学生人数如表 2－6 所示，各村庄间的距离如图 2-21 所示. 现在计划建造一所医院和一所小学，问医院应建在哪个村庄才能使最远村庄的人到医院看病所走的路最短? 又问小学建在哪个村庄使得所有学生上学走的总路程最短?

表 2－6

村庄	v_1	v_2	v_3	v_4	v_5	v_6
小学生	50	40	60	20	70	90

3. 某建筑公司签订了一项合同，要为一家制造公司建造一座新的加工厂. 合同规定工厂的完工期限为 12 个月. 要是工厂不能在一年内完工，就要赔款，因此建筑公司通过认真分析，找出建筑工厂必须完成的各道工序和这些工序之间的先后关系，并估计出它们延续的时间，如表 2－7 所示. 建立图论模型，并进行分析，为建筑公司制定工程完工计划提供理论依据.

表 2－7　建筑工序表

工序(事项)	估计周数	紧前事项
1　平整土地	4	无
2　打桩	1	1
3　运进钢材	3	无
4　运进混凝土	2	无
5　运进木料	2	无
6　运进水管和电器材料	1	无
7　浇注地基	7	2,3,4
8　焊接钢梁	15	3,7
9　安装生产设备	5	7,8
10　分隔办公室	10	5,7,8
11　安装水电和电器	11	6,8,10
12　装饰墙壁	5	8,10,11

4. 设有 A、B、C 三只水桶，容积分别为 12 升，9 升和 5 升. A 中盛满 12 升水. 试问：如何利用空桶 B 和 C 来把 A 中的水分成两半？

5. 非平凡连通图 G 是 Euler 图当且仅当 $E(G)$ 能划分为不交子集 $E_i(1\leqslant i\leqslant n)$ 的并，且对任意 $1\leqslant i\leqslant n$，E_i 的导出子图是 G 的一个圈.

6. 图 2-22 是奥伦治的威廉王的迷宫平面图（建立于 1690 年，至今仍屹立着）. 试画出该迷宫所对应的图，再找一条从进口 x 到中心 y 的行走路线. 怎样确定从进口 x 到中心 y 的最佳行走方案.

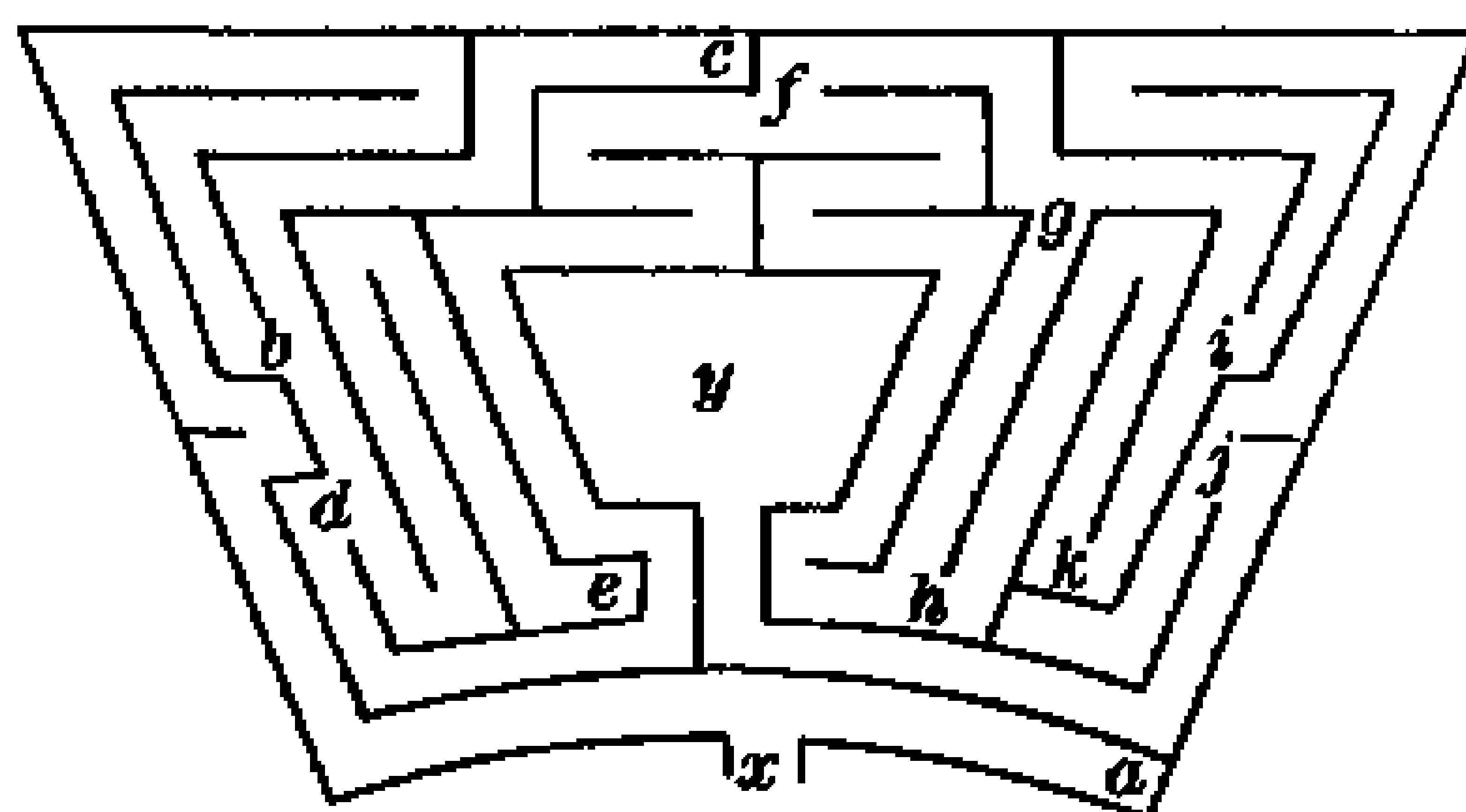


图 2-22

7. 给定一个由 16 条线段构成的图形，如图 2-23 所示. 证明：不能引一条折线与每一线段恰好相交一次.（折线可以是不封闭的和自由相交的，但它的顶点不在给定的线段上，而边也不通过线段的公共端点）.

8. 某研究所收到由 n 个人所寄的一些问题的解，他们发现每个人寄来 4 个不同问题的解，每个问题的解恰好由两个人同时给出. 问他们共收到几个不同问题的解，并证明研究所可以分二次发表这些问题的解，使每人每次恰好被提到两次.

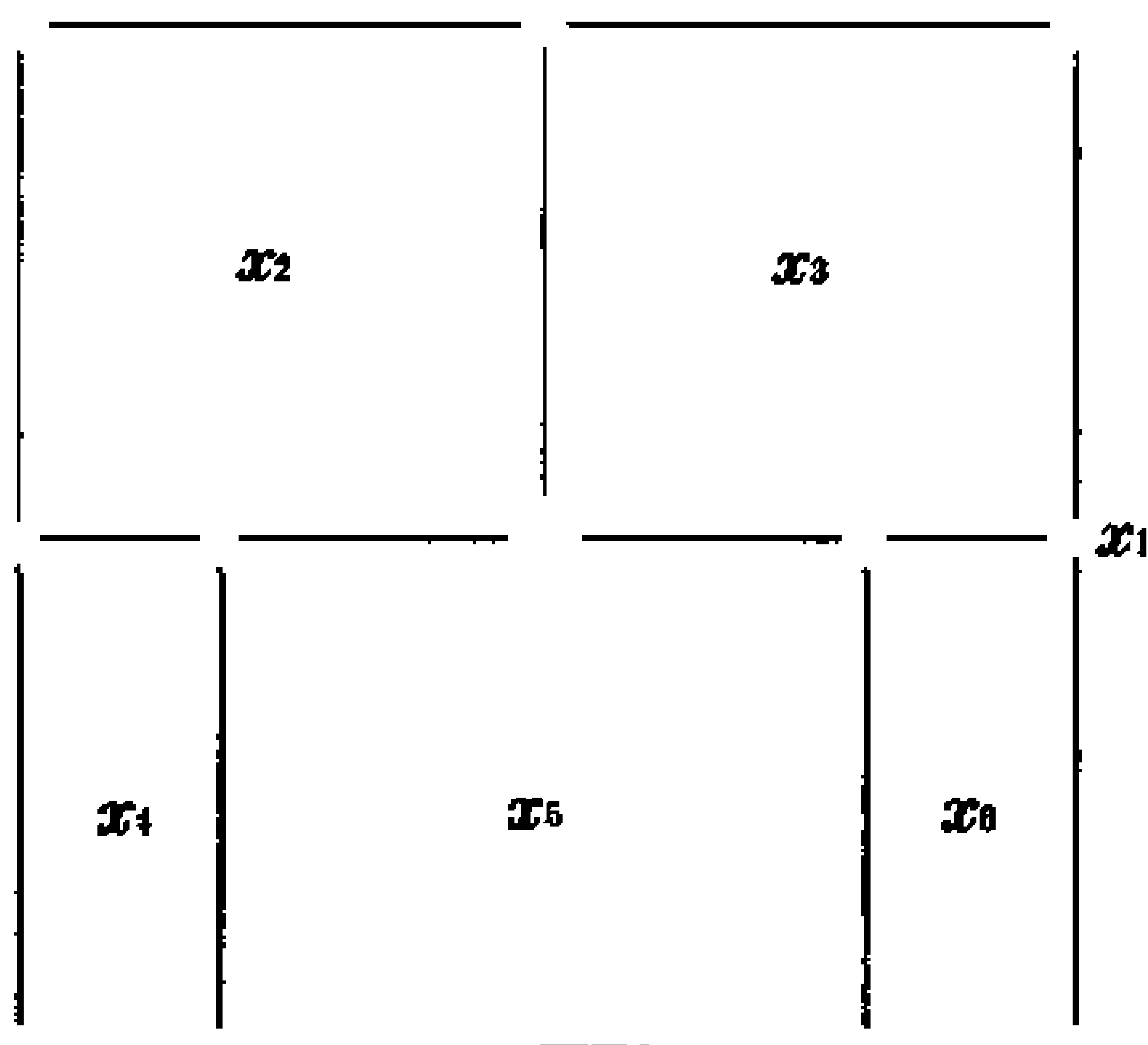


图 2-23

9. 设 G 是具有 n 个结点的简单无向图，如果 G 中每一对结点度数之和大于等于 $n-1$，则 G 中存在一条汉密尔顿路；若 G 中每一对结点度数之和大于等于 n，则 G 中的存在一条汉密尔顿回路.

10. 某次会议有 20 人参加，其中每个人都至少有 10 个朋友，这 20 人围一圆桌入席，要想使与每个人相邻的两位都是朋友是否可能？

11. 已知 a,b,c,d,e,f,g 七个人中，a 会讲英语，b 会讲英语和汉语；c 会讲英语、意大利语和俄语；d 会讲汉语和日语；e 会讲意大利语和德语；f 会讲俄语、日语和法语；g 会讲德语和法语. 能否将他们的座位安排在圆桌旁，使得每个人都能与他身边的人交谈.

12. 今将 6 人分成 3 组（每组 2 人）去完成 3 项任务. 已知每个人至少与其余 5 个人中的 3 个人能相互合作. 问：

(1)能否使得每组的 2 个人都能相互合作？

(2)你能给出几种不同的分配方案？

13. 找一种 9 个 a,9 个 b,9 个 c 的圆形排列,使由字母 $\{a,b,c\}$ 组成的长度为 3 的 27 个字的每个字仅出现一次.

14. 设 G 是分划为 X,Y 的二分图,且 $|X|\neq|Y|$,则 G 一定不是哈密顿图.

15. 设简单图 $G=\langle V,E\rangle$, $|V|=n$, $|E|=m$,若有 $m\geqslant C_{n-1}^2+2$,则 G 是哈密顿图.

16. 有一个 $3\times3\times3$ 的立方体,它是由 27 个 $1\times1\times1$ 的子立方体组成.两个子立方体是相邻的,当且仅当它们具有公共面.一根导线,从八个角的某一个子立方体出发,穿过公共面进入其相邻的子立方体.问:最后能否穿过所有子立方体而终止于该立方体的中心?

17. n 个人参加一个会议,在会议期间,每天都要在一只圆桌上共进晚餐,如果要求每次晚餐就座时,每个人相邻就座者都不相同,问这样的晚餐最多能进行多少次?

18. 对下列各函数给出一个尽可能好的大 O 估计.

(1) $(n^2+8)(n+1)$

(2) $(n\log n+n^2)(n^3+2)$

(3) $(n!+2^n)(n^3+\log(n^2+1))$

(4) $(n^2+n(\log n)^3)(2^n+n^3)$.

19. 求证 $x^2+4x+17$ 是 $O(x^3)$,但 x^3 不是 $O(x^2+4x+17)$.

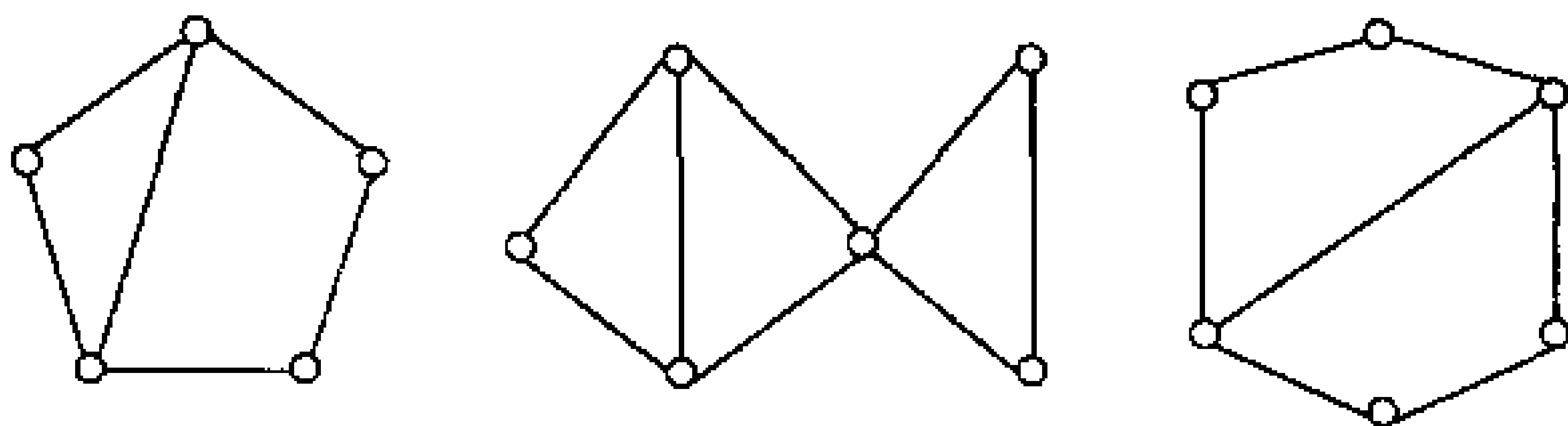


图 2-24

20. 设 k 为正整数，证明 $1^k + 2^k + \cdots + n^k$ 是 $O(n^{k+1})$.

21. 构造一种格雷码，其中码字都是长度为 6 的位串.

22. 求图 2-24 中各图的闭包.

23. 画出求加权图 G 的任意两点间最短路的 Warshall 算法框图，并编程上机运行.

24. 显示算法，判断图 G 是否连通，若不连通，有几个连通分支，要求画出算法框图，再编程上机运行.

第三章　树及其应用

第一节　树的基本概念

树的术语起源于植物学和家谱学. 早在1857年,英国数学家亚瑟·凯来(Arthur Cay ley,1821~1895)就发现了树,当时他正在试图列举形为C_nH_{2n+2}的化合物的同分异构体. 树具有广泛的应用,特别是在计算机科学和管理科学中. 如用树构造存储和传输数据的有效编码,用树构造最便宜的电话线连接分布式计算机网络,用树模拟一系列决策完成的过程等.

定义1　**树**(**tree**)是无圈连通无向图. 树中度数为1的结点称为树的**叶**(**leafage**). 树中度数大于1的结点称为树的**分枝点**(**branch vertex**)或内点. 不相交的若干树称为**森林**(**forest**),即森林的每个连通分支是树.

定理1　T是树$\Leftrightarrow T$中无环,且任何不同两顶点间有且仅有一条路.

证明　($\Rightarrow$)$\forall x,y\in V(T),x\neq y$. 由于$T$是连通图,则$x$与$y$之间一定有路. 设$P_1$和$P_2$是$T$中两条不同的$xy$路,则$P_1\cup P_2$是闭链,且存在边$e\in P_1$,但$e\notin P_2$. 显然子图$P_1\cup P_2-e$是连通图. 设$e=\{u,v\}$,则子图$P_1\cup P_2-e$中存在$uv$路$P$,于是$P+e$是圈. 这与已知矛盾. 因此树$T$的任何不同两顶点间有且仅有一条路.

($\Leftarrow$)假设T中含圈C. 若$V(c)=1$,则T中有环;若$V(c)=2$,则T中有平行边;若$V(c)\geqslant 3$,$\forall x,y\in V(c)$,有两条不同的

xy 路. 这些都与已知矛盾,因此 T 是无圈连通图,即 T 为树.

定理 2　T 是树$\Leftrightarrow$　T 连通且对 $\forall e\in E(T)$,有 $W(T-e)=2$.

证明　($\Rightarrow$)已知 T 是树,由定理 1 知 T 是简单连通图.

设 $e=\{x,y\}\in E(T)$,则 $W(T-e)\leqslant 2$. 另一方面,由定理 1 知 xey 是 T 中唯一 xy 路,所以 x 和 y 在 $T-e$ 的不同连通分支,于是 $W(T-e)\geqslant 2$. 故 $W(T-e)=2$.

($\Leftarrow$)假设 T 含圈 C,并设 $e=\{x,y\}\in E(C)$. 由于 T 连通且 $W(T-e)=2$,所以 e 不是环,即 x 和 y 在 $T-e$ 的不同连通分支中. 然而 $C-e$ 是 $T-e$ 中 xy 路,矛盾,所以 T 是树.

定理 3　T 是树$\Leftrightarrow T$ 连通且 $\varepsilon=n-1$,其中 ε 为 T 的边数,n 为 T 的顶点数.

证明　($\Rightarrow$)设 T 的边数为 $\varepsilon(T)$,顶点数为 $V(T)$. $\varepsilon(T)=1$, $V(T)=2$ 时,显然 $\varepsilon(T)=V(T)-1$. 现假设对所有边数小于 $\varepsilon(T)$ 的树,式 $\varepsilon(T)=V(T)-1$ 成立.

$\forall e\in E(T)$,由定理 2 知,$T-e$ 有两个连通分支,设其为 T_1 和 T_2,则 $\varepsilon(T_1)=V(T_1)-1$, $\varepsilon(T_2)=V(T_2)-1$. 而 $\varepsilon(T)=\varepsilon(T_1)+\varepsilon(T_2)+1=V(T_1)+V(T_2)-1=V(T)-1$. 即 $\varepsilon=n-1$.

($\Leftarrow$)$n=1$ 时 $\varepsilon=0$; $n=2$ 时, $\varepsilon=2$; 显然成立. 设任何 n 阶且边数 $\varepsilon=n-1$ 的连通图不含圈.

对于 $n+1$ 阶且边数 $\varepsilon=n$ 的连通图 T,有 $\delta(T)\geqslant 1$. 若 $\delta(T)\geqslant 2$,则 $2\varepsilon=2n=\sum\limits_{x\in V(T)}\deg(x)\geqslant 2(n+1)$,这是不可能的. 所以存在 $x\in V(T)$,使 $\deg(x)=1$. 于是 $T-x$ 是 n 阶图,且 $T-x$ 只有 $n-1$ 条边. 由归纳假设知 $T-x$ 不含圈,从而得 T 是不含圈的连通图. 即 T 为树.

推论　T 是林$\Leftrightarrow$　$\varepsilon=n-w$,其中 ε 为 T 的边数,n 为 T 的

顶点数，w 为 T 的连通分支数.

定理 4 设 T 是 n 阶非平凡树(平凡图称为平凡树)，则 T 中至少有 2 片树叶.

证明 设 T 是 n 阶非平凡树，则 $\delta(T)\geqslant 1$. 并设 T 有 ε 条边，k 片树叶，则 $2\varepsilon=\sum_{i=1}^{n}\deg(v_i)\geqslant k+2(n-k)=2n-k$，而 $\varepsilon=n-1$，故 $k\geqslant 2$.

定义 2 设 T 是有向图，若 T 的基础图是树，称 T 是**有向树**(**directed tree**).

定义 3 仅一个结点的入度为 0，其余所有结点的入度都为 1 的有向树称为**根树**(**rooted tree**). 入度为 0 的结点称为**根**(**root**). 出度为 0 的结点(即度数为 1 的结点)仍称为叶；出度不为 0 的结点称为**分枝点**或**内点**(根是特殊的分枝点除非它是图中唯一的顶点).

由根到某一顶点 v 的有向路的长度，称为顶点 v 的**层数**(**level**). 根树的**高度**就是顶点层数的最大值.

例 1 用根树可以表示家庭之间的关系.

设某祖先 a 有两个儿子 b,c，b 与 c 又分别有三个儿子 d,e,f 和 g,h,i，而 d 有一个儿子 j，h 有两个儿子 k,l. 这样的家庭关系可以用如下的根树表示，如图 3-1 所示.

因此，根树也称家族树. 同时也有一些类似的术语. 比如，设 u 是根树的分枝点，若从 u 到 v 有一条有向边 $\langle u,v\rangle$，则可称 v 为 u 的儿子或 u 为 v 的父亲；若一个分枝点有两个儿子，则称它们为兄弟. 若从 u 到 z 有一条有向路，则可称 z 为 u 的后代或 u 为 z 的祖先等等.

根树概念非常重要，原因在于它描述了一个离散结构的层次关系，而层次结构是一种重要的数据结构，因而有着广泛的应用.

在画根树时，把根画在图的顶端. 指标根树中边的方向的箭头

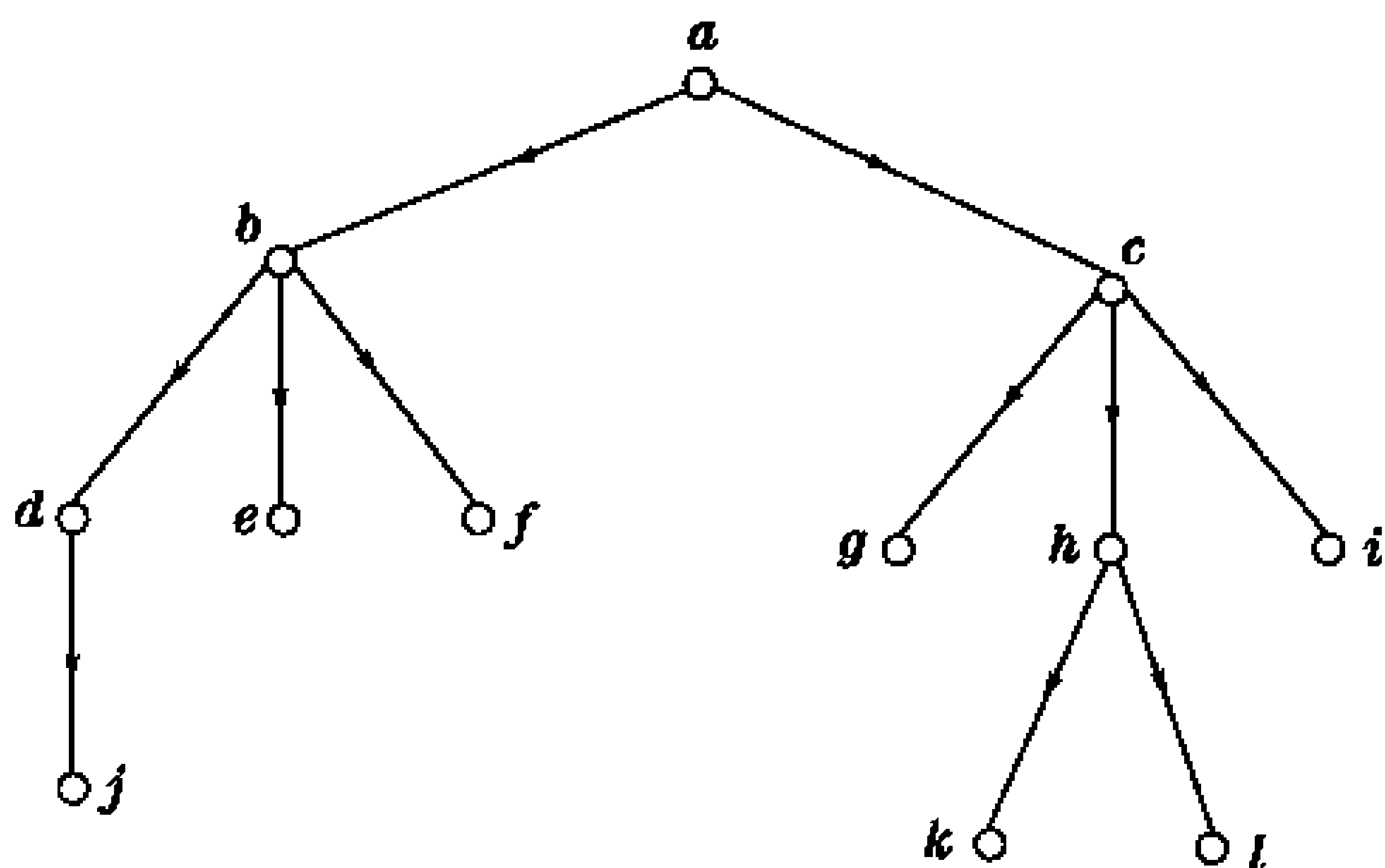


图 3-1

可以省略,因为对根的选择确定了边的方向.

若 a 是根树里的顶点,则以 a 为根的**子树**是由 a 和 a 的后代以及这些顶点所关联的边组成的该树的子图.

定义 4 如果在根树中规定了每一层上的顶点的次序,这样的根树称为**有序树**(**ordered tree**).

一般地,在画出的有序树中,规定同一层次的顶点的次序为从左至右(即不能互换).也可以规定用边的次序来代替顶点的次序.

如图 3-2 所示的(a),(b)是两棵不同的有序树.

例如,用图 3-3 所示的有序树表示代数式 $v_1v_2+v_3(v_4+\frac{v_5}{v_6})$.

树中的叶表示参加运算的元素,分枝点表示相应的运算.运算的次序是先左后右,自下而上.在分枝点上的运算指示,只有在其左右子树均已变成数或式子时才进行.

定义 5 每个结点的出度不大于 m 的有序根树称为 $\boldsymbol{m}$ **叉树**.每个结点的出度恰好为 m 或零的 m 叉树称为**完全** $\boldsymbol{m}$ **叉树**.所有树叶层次相同的完全 m 叉树称为**正则** $\boldsymbol{m}$ **叉树**.

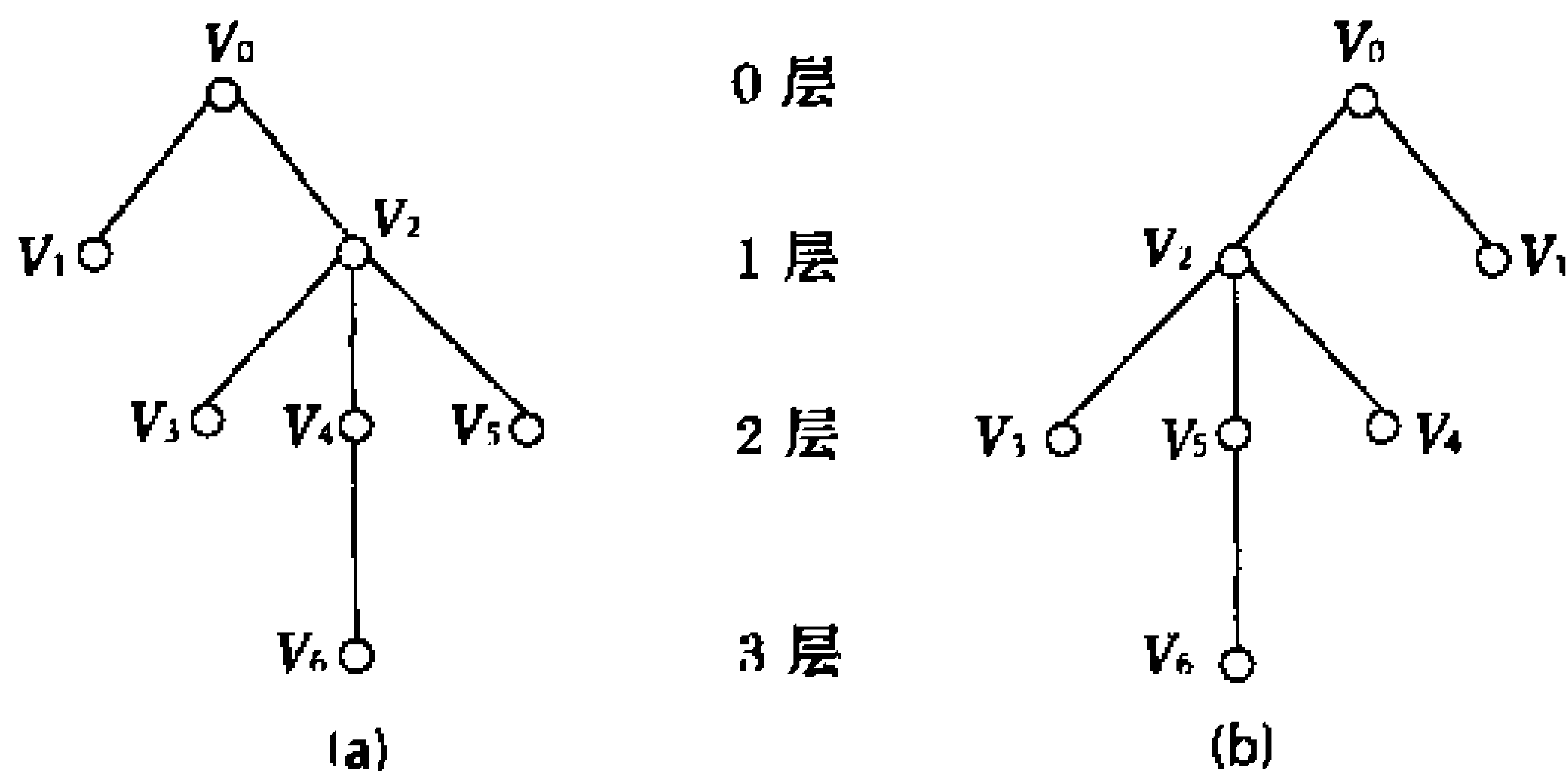


图 3-2

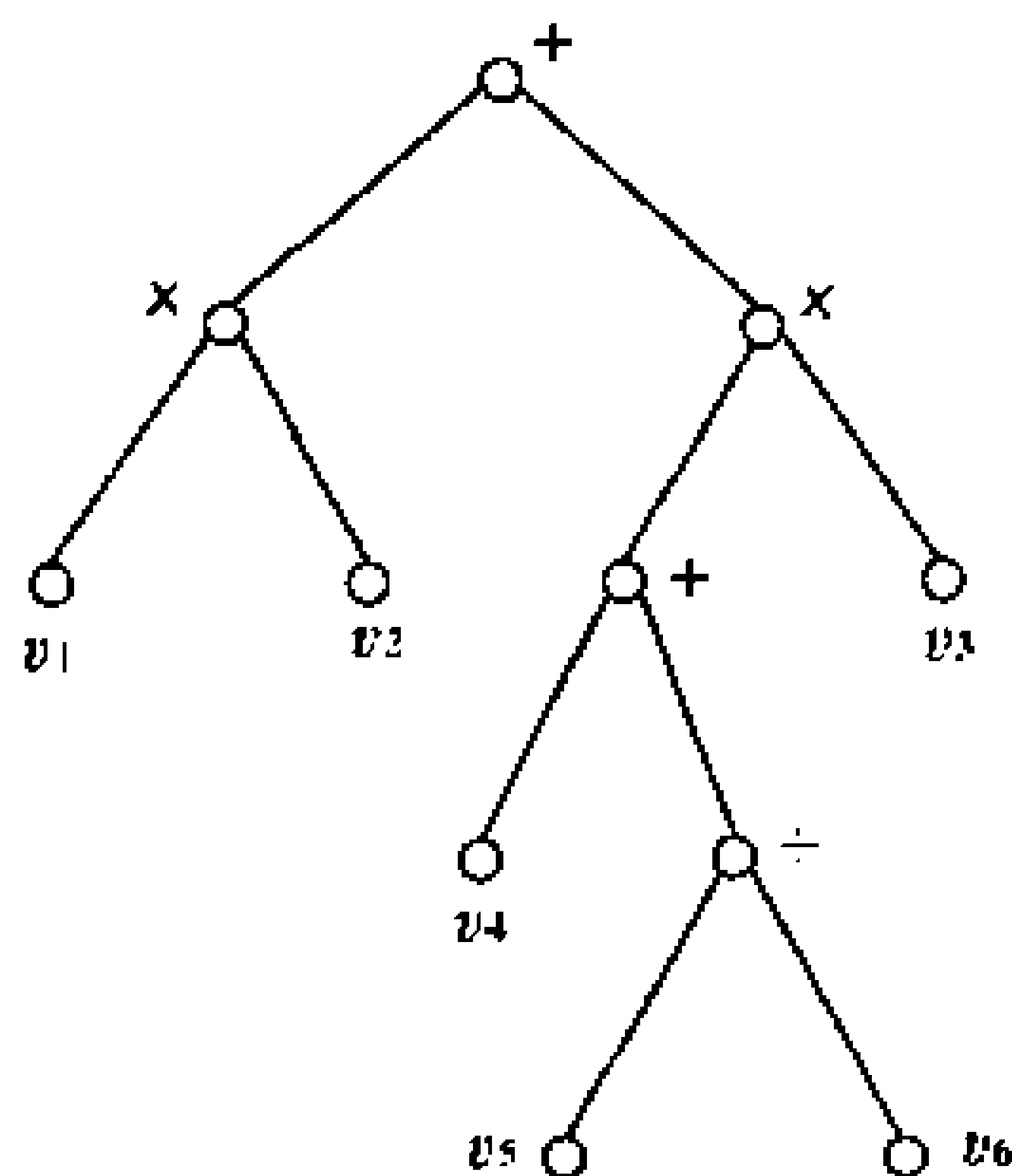


图 3-3

定理 5 设 T 是完全 m 叉树，且 T 的树叶数为 t，分枝点数为 i，则 $(m-1)i=t-1$.

证明 在完全 m 叉树中，每片叶子的度数为 1，根的度数为 m，其余每个分枝点的度数为 $m+1$. 并设 T 有 ε 条边，则 $2\varepsilon=t+m+(i-1)(m+1)$，又因为 $\varepsilon=t+i-1$，故 $(m-1)i=t-1$. 证毕

下面，我们引进图的离径、半径与中心的概念.

定义 6 图 G 的一个顶点 v 的**离径** $R(v)$ 定义为

$$R(v)=\max_{u\in V(G)}\{d(v,u)\}$$

图 G 的**半径** $R(G)$ 定义为

$$R(G)=\min_{v\in V(G)}\{R(v)\}$$

所有满足 $R(v)=R(G)$ 的顶点 v 都称作 G 的**中心**.

显然,一个图 G 的直径为 $\max_{v\in V(G)}\{R(v)\}$.

例如,图 3-4 所示的图 G 中, $R(u_1)=R(u_4)=R(u_6)=3$, $R(u_2)=R(u_3)=R(u_5)=2$. 故 G 的直径为 3,半径为 2, u_2, u_5 与 u_3 是 G 的中心.

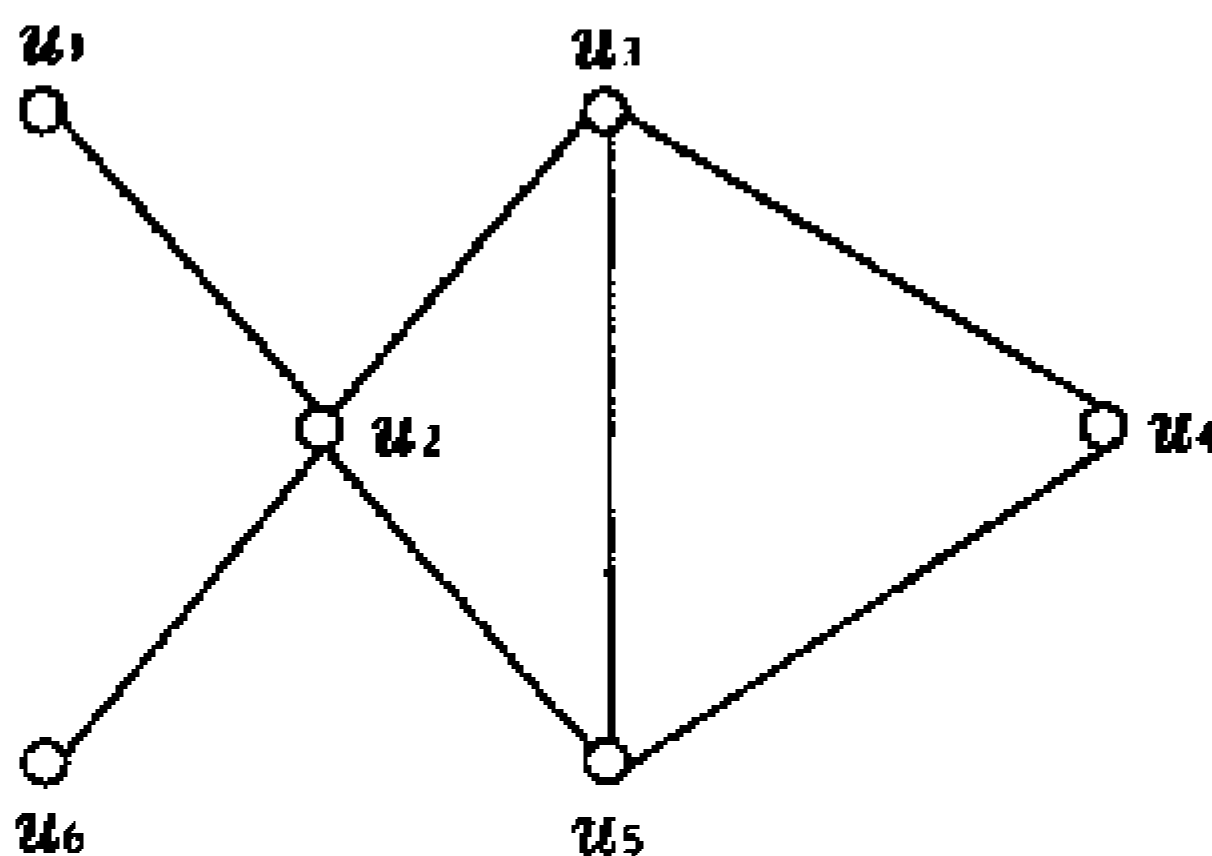


图 3-4　G

定理 6 设 $P=u_1u_2\cdots u_lu_{l+1}$ 是树 T 的一条最长路,则.

(1) T 的直径为 l;

(2)若 l 为奇数,设 $l=2k-1$,那么 T 的半径为 k, T 有两个相邻的中心,即为 u_k 和 u_{k+1}. 并且每一条长为 l 的路都通过这两个中心;

(3)若 l 为偶数,设 $l=2k$,那么 T 的半径为 k, T 中只有一个中心,即为 u_{k+1}. 并且每一条长为 l 的路都通过中心 u_{k+1}.

证明 (1)由于树 T 中只有唯一的一条 u_1-u_{l+1} 路 P,故 $d(u_1,u_{l+1})=l$ 即为 T 的半径.

(2)设 u 是 T 中的任意一个顶点,则

$d(u,u_1)+d(u,u_{2k})=d(u_1,u)+d(u,u_{2k})\geqslant d(u_1,u_{2k})=2k-1$,所以 $d(u,u_1)\geqslant k$ 或 $d(u,u_{2k})\geqslant k$,因此 u 的离径 $R(u)\geqslant k$.所以 T 的半径 $R(T)\geqslant k$.

对于 T 中任意一个顶点 u,如果 u 在 P 上,则显然有 $d(u,u_k)\leqslant k$.如果 u 不在 P 上,则 P 中必存在一个顶点 $u_i(2\leqslant i\leqslant 2k-1=l)$,$T$ 中有一条连接 u 与 u_i 的路 Q,使 $u_1,u_2,\cdots,u_{i-1},u_{i+1},\cdots,u_{2k}$ 都不在 Q 上.

由于 P 是 T 的最长路,所以有以下两个不等式:

$$d(u,u_i)\leqslant i-1, d(u,u_i)\leqslant l+1-i$$

如果 $i\leqslant k$,由于 $d(u_i,u_k)=k-i$,故

$$d(u,u_k)=d(u,u_i)+d(u_i,u_k)\leqslant i-1+k-i=k-1<k$$

如果 $i>k$,则 $d(u,u_k)=d(u,u_i)+d(u_i,u_k)\leqslant l+1-i+i-k=l+1-k=k$,从而对 T 的每个顶点 u,$d(u,u_k)\leqslant k$,因此 $R(u_k)\leqslant k$,从而 $R(T)\leqslant k$

因此,$R(T)=k$,且 $R(u_k)=k$,即 u_k 为 T 的一个中心.同理可证 u_{k+1} 也是 T 的一个中心.

下面证明长为 l 的第一条路 P' 一定通过中心 u_k 与 u_{k+1}.设 P' 的两个端点分别是 u 和 u',则 u 和 u' 均不在 P 上.在 P 上有两个顶点 u_i 和 $u_j(2\leqslant i,j\leqslant 2k-1)$,$T$ 中存在 $u-u_i$ 路 Q_1 和 $u'-u_j$ 路 Q_2,使 $(V(Q_1)-\{u_i\})\cap V(P)=\emptyset$,$(V(Q_2)-\{u_j\})\cap V(P)=\emptyset$.不妨设 $i\leqslant j$.因为 T 是树,故 $P'=Q_1\cup P(u_i,u_j)\cup Q_2$

如果 $i\leqslant j\leqslant k$,则

$$\begin{aligned}d(u,u')&\leqslant d(u,u_i)+d(u_i,u_j)+d(u_j,u')\\&\leqslant(i-1)+(j-i)+(j-1)\\&=2(j-1)\\&\leqslant 2(k-1)<2k-1=l\end{aligned}$$

同理,若 $k<i\leqslant j$,则 $d(u,u')<l$.这些都与 P' 是长为 l 的 $u-u'$ 路相矛盾.因此 $i\leqslant k<j$.这就证明了长为 l 的路一定通过 T 的中心 u_k 和 u_{k+1}.

(3)的证明类似于(2).

例 2 平面上有 $n(n\geqslant 3)$条线段,其中任意 3 条都有公共端点. 证明这 n 条线段有一个公共端点.

证明 把这 n 条线段的端点视为一个图 G 的顶点,线段为 G 的边.由题意知,G 是无回路的连通图,故 G 是树,且 G 的直径为 2,$|V(G)|=|E(G)|+1=n+1$.取 $u\in V(G)$,使 $d_G(u)=\Delta(G)$,则 $d_G(u)\geqslant 2$.若存在 $v\in V(G)-\{u\}$,$d_G(v)\geqslant 2$.设 $x\in N_G(u)-\{v\}$,$y\in N_G(v)-\{u\}$,如果边$\{u,v\}\in E(G)$,因 G 是树,$x\neq y$ 且 $d(x,y)=3$ 与 G 的直径为 2 矛盾;如果边$\{u,v\}\notin E(G)$,则存在 $x_1\in N_G(u)-\{x\}$,$y_1\in N_G(v)-\{y\}$,此时 G 的三条边$\{x_1,u\}$,$\{u,x\}$,$\{v,y_1\}$无公共端点,与题设矛盾.故 G 中只有一个顶点 u 是非悬挂点,而其余 n 个顶点是悬挂点,则

$$2n=2|E(G)|=\sum_{x\in V(G)}d(x)=d_G(u)+n$$

所以 $d_G(u)=n$,因为 u 就是这 n 条边的公共端点,也即为 n 条线段的公共端点.

第二节　支撑树的计数

定义 1 如果 T 是 G 的一个生成子图而且又是一棵树,则称 T 是图 G 的一棵**生成树**(spanning tree)或**支撑树**.生成树 T 中的边称为树 T 的**树枝**(branch),不在生成树 T 中的 G 的边,称为树 T 的**弦**(chord).

定理 1 图 G 有生成树$\Leftrightarrow G$ 为连通图.

证明 ($\Rightarrow$)如果 G 不连通,那么它的生成子图都不连通.因

而 G 不能有生成树.

($\Leftarrow$)已知 G 连通.设 T 是 G 的边数最少的连通生成子图.于是对 $\forall e\in E(T)$,$T-e$ 不是连通图,即 T 的每一边都是割边,因此 T 是一棵树,而 T 又是 G 的生成子图,因而 T 是 G 的一棵生成树.证毕

下面我们来讨论生成树的计数问题.

定义 2 设 D 是无环有向图,且 D 有 n 个结点 ε 条边.在 D 的关联矩阵 $M_{n\times\varepsilon}$ 中划去任意结点 x 所对应的行,得到一个 $(n-1)\times\varepsilon$ 阶矩阵 M_x,称 M_x 为 D 的一个**基本关联矩阵**.

定理 2 (**Binet-Cauchy**)已知两个矩阵 $A=(a_{ij})_{m\times n}$,$B=(b_{ij})_{n\times m}$.若 $m\leqslant n$,则 $\det(AB)=\sum_i A_iB_i$,其中 A_i 和 B_i 都是 m 阶行列式,A_i 是从 A 中取不同的 m 列所成的行列式,B_i 是从 B 中取相应的 m 行所构成的行列式.

例如 已知 $A=\begin{bmatrix}4 & 3 & 2\\ -2 & 4 & 3\end{bmatrix}$ $B=\begin{bmatrix}5 & 1\\ 0 & 3\\ 4 & 2\end{bmatrix}$

$$AB=\begin{bmatrix}28 & 17\\ 2 & 16\end{bmatrix}\qquad \det(AB)=414$$

另一方面,有

$$\det(AB)=\begin{vmatrix}4 & 3\\ -2 & 4\end{vmatrix}\begin{vmatrix}5 & 1\\ 0 & 3\end{vmatrix}+\begin{vmatrix}4 & 2\\ -2 & 3\end{vmatrix}\begin{vmatrix}5 & 1\\ 4 & 2\end{vmatrix}+\begin{vmatrix}3 & 2\\ 4 & 3\end{vmatrix}\begin{vmatrix}0 & 3\\ 4 & 2\end{vmatrix}=414.$$

定理 3 设 B_k 是有向弱连通图 D 的某个基本关联矩阵,则 D 的不同支撑树的数目是 $\det(B_kB_k^T)$.其中 B_k^T 为 B_k 的转置矩阵.

例 1 图 D 如图 3-5 所示.解答下列问题:

(1)求 D 的不同生成树的数目.

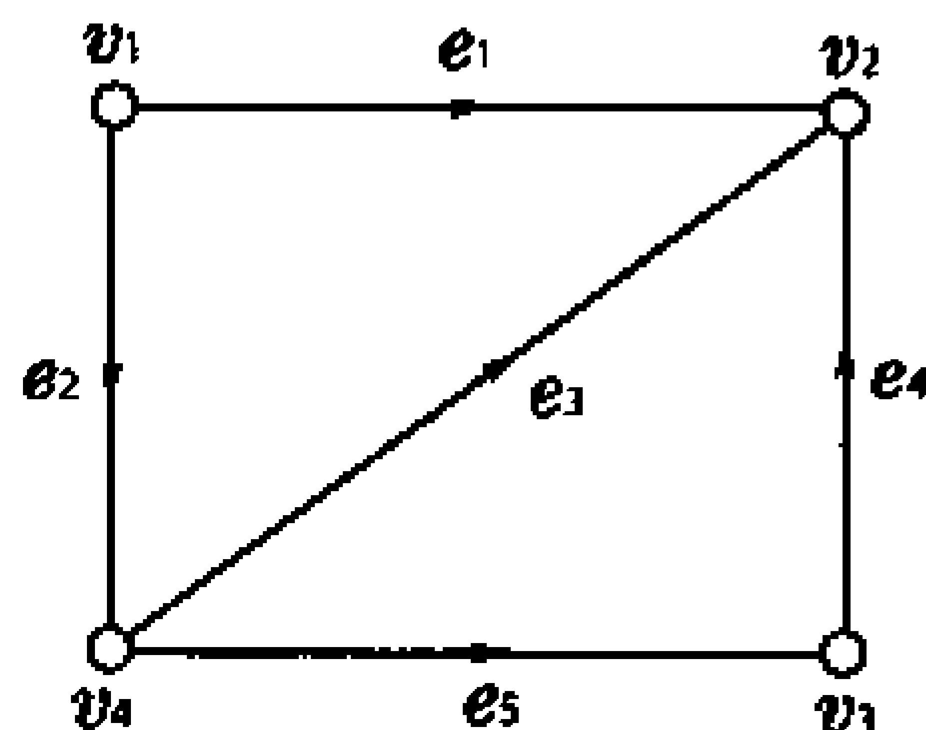


图 3-5

(2)求 D 的不含边 e_4 的不同生成树的数目.

(3)求 D 的必含边 e_3 的不同生成树的数目.

解 (1)D 的关联矩阵 M 为

$$M=\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

选 v_4 的基本关联矩阵,并记作 B_4,则

$$B_4=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

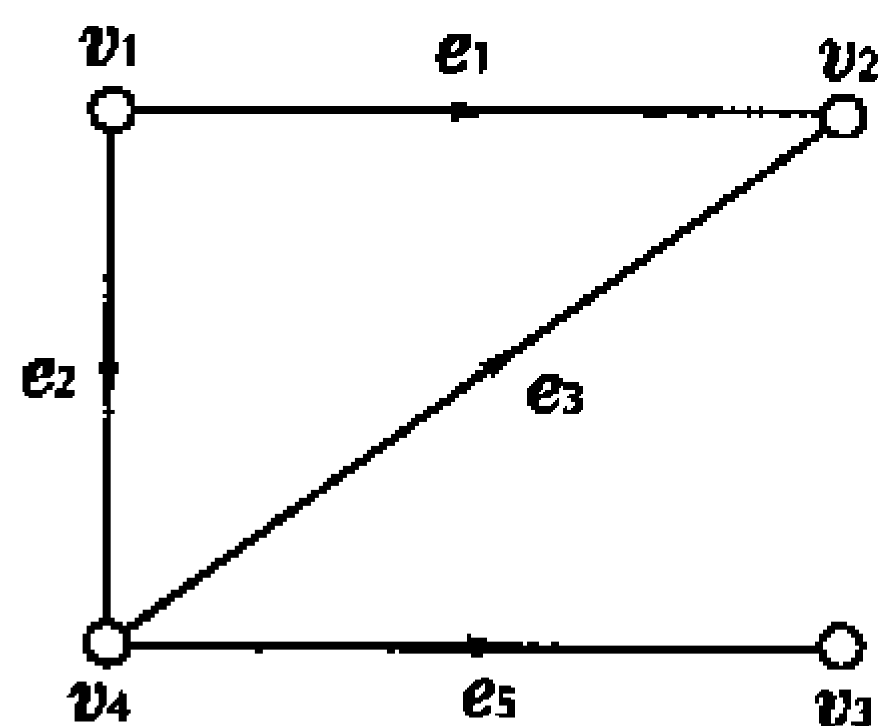


图 3-6

于是 $$\det(B_4B_4^T)=\det\begin{bmatrix}2 & -1 & 0\\ -1 & 3 & -1\\ 0 & -1 & 2\end{bmatrix}=8$$

所以,D 中有 8 棵不同的生成树.

(2)为求图 D 中不含边 e_4 的生成树的数目,转化为求图 3-6 所示图 D_1 的不同生成树的数目.图 D_1 的关联矩阵为 M_1

$$M_1=\begin{array}{c}\\ v_1\\ v_2\\ v_3\\ v_4\end{array}\begin{array}{c}\begin{matrix}e_1 & e_2 & e_3 & e_5\end{matrix}\\ \begin{bmatrix}1 & 1 & 0 & 0\\ -1 & 0 & -1 & 0\\ 0 & 0 & 0 & -1\\ 0 & -1 & 1 & 1\end{bmatrix}\end{array}$$

选 v_4 的基本关联矩阵,记作 C_4,则

$$C_4=\begin{bmatrix}1 & 1 & 0 & 0\\ -1 & 0 & -1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$$

于是 $\det(C_4C_4^T)=\det\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & 0\\ 0 & 0 & 1\end{pmatrix}=5.$

所以,图 D 中不含边 e_4 的支撑树的数目为 5.

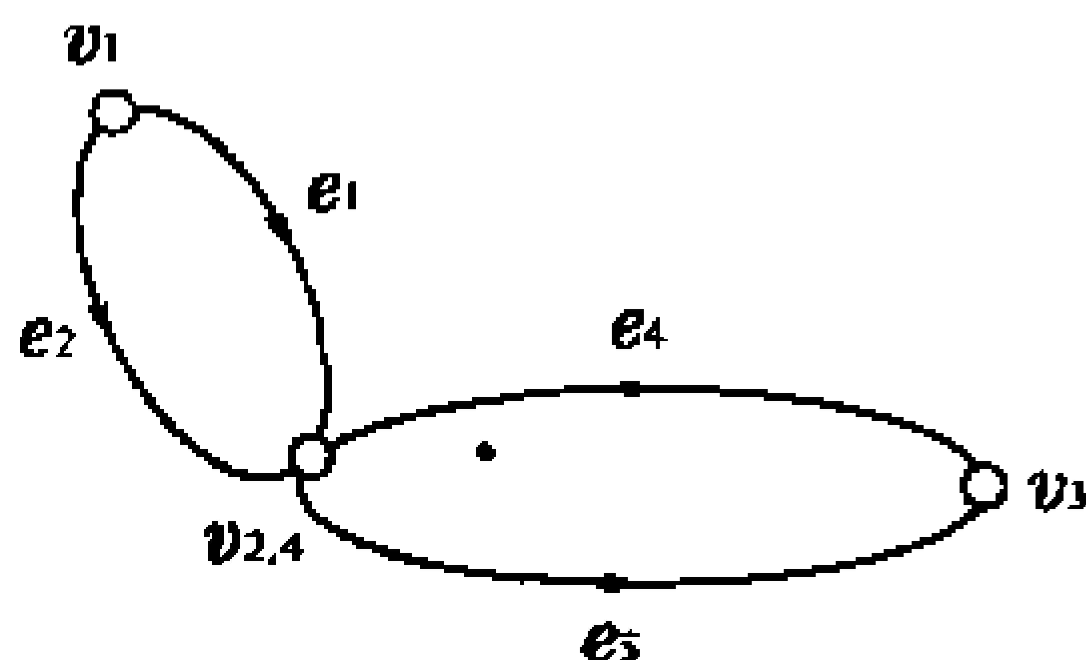


图 3-7

(3)为求图 D 中必含边 e_3 的支撑树的数目，转化为求图 3-7 所示 D_2 的不同生成树数目. 图 D_2 的关联矩阵为 M_2

$$M_2=\begin{matrix} & e_1 & e_2 & e_4 & e_5 \\ v_1 & -1 & -1 & 0 & 0 \\ v_{2,4} & -1 & -1 & -1 & 1 \\ v_3 & 0 & 0 & 1 & -1 \end{matrix}$$

取 $v_{2,4}$ 的基本关联矩阵，记作 B，则 $B=\begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$

于是 $\det(BB^T)=\det\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}=4.$

所以，图 D 中必含边 e_3 的不同支撑树有 4 棵.

说明 (1)记 $C=B_kB_k^T=(C_{ij})\quad i\neq k, j\neq k$，则

$$C_{ij}=\begin{cases} d_D(v_i) & i=j \\ -\text{关联 } v_i \text{ 与 } v_j \text{ 的边数} & i\neq j \end{cases}$$

(2)对于连通无向图 G 的支撑树计数，只需考虑 G 的定向图.

下面来考虑生成根树的计数. 为此，先看一个实际问题. 如图 3-8(a)所示根树 T 的关联矩阵记为 M，v_0 的基本关联矩阵记为 B_0，则

$$M=\begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ v_0 & 1 & 1 & 0 & 0 \\ v_1 & -1 & 0 & 1 & 1 \\ v_2 & 0 & 0 & -1 & 0 \\ v_3 & 0 & 0 & 0 & -1 \\ v_4 & 0 & -1 & 0 & 0 \end{matrix},\quad B_0=\begin{bmatrix} -1 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

现对根树 T 的结点和边重新编号，使每边 $e=\langle v_i, v_j\rangle$ 都满足 v_i 的编号小于 v_j 的编号，同时 $e=\langle v_i, v_j\rangle$ 的编号为 e_j；这样得到

如图 3-8(b)所示根树 T'.

树 T'的结点 v_0 的基本关联矩阵

$$B_0' = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

这样 B_0' 是对角矩阵,且 B_0' 可以由 B_0 通过初等变换得到.显然

$$\det(B_0) = \det(B_0) = \det \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

当然,只有有向树才有此特征.

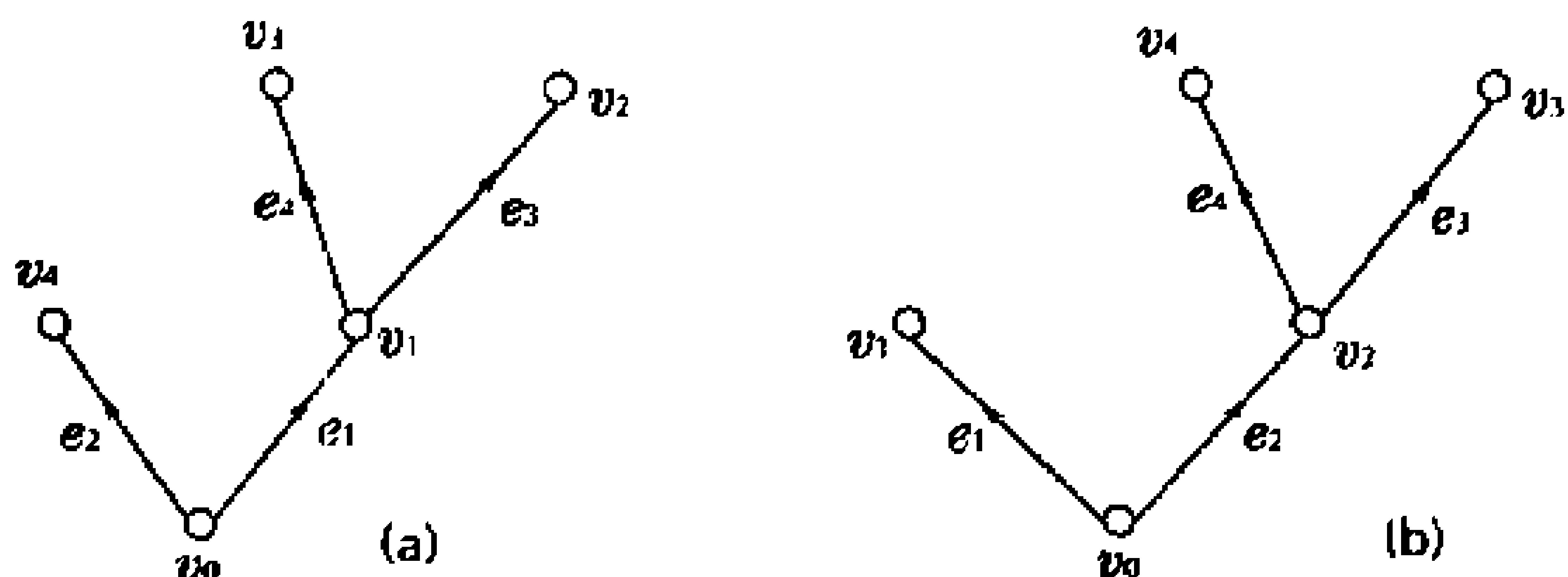


图 3-8

定理 4 有向弱连通图 D 中以 v_k 为根的不同支撑根树数目为 $\det(\vec{B}_k\vec{B}_k^T)$,其中 $\vec{B}_k$ 表示D 的 v_k 的基本关联矩阵B_k 中将全部 1 元素改为 0 元素之后的矩阵.

第三节　深度优先搜索与广度优先搜索

求连通简单图 G 的一棵生成树的许多方法中，**深度优先搜索（depth first search）**是一个十分重要的算法．而且它在其它许多方面也得到了广泛的应用，是解决许多实际问题的有力工具．

深度优先搜索的基本思想：任意选择图 G 的一个顶点 v_0 作为根．通过相继地添加边来形成在顶点 v_0 开始的路，其中每条新边都与路上的最后一个顶点以及不在路上的一个顶点相关联．继续尽可能多地添加边到这条路．若这条路经过图 G 的所有顶点，则这条路即为 G 的一棵生成树．若这条路没有经过图 G 的所有顶点，不妨设形成这条路的顶点顺序为 $v_0,v_1,\cdots\cdots,v_n$．则返回到路里的次最后顶点 v_{n-1}，若有可能，则形成在顶点 v_{n-1} 开始的经过还没有访问过的顶点的路；否则，则返回到路里的顶点 v_{n-2}，然后再试．重复这个过程，在所访问过的最后一个顶点上开始，在路上一次返回一个顶点，只要有可能就形成新的路，直到不能添加更多的边为止．因为图 G 是有有限边数的连通图，所以这个过程以产生生成树而告终．在这个算法的一个阶段上，路的末端顶点将是根树的叶，而在其上开始构造路的顶点是根树的分枝点．

深度优先搜索也称为**回溯（back tracking）**．

例 1　用深度优先搜索来找出图 3-9 所示图 G 的生成树．

解　任意地从顶点 d 开始．相继地添加与还不在路上的顶点相关联的边，只要有可能就继续下去，产生路 d,e,f,g,h,j,k．下一步，回溯到 j，不存在从 j 开始的包含还没有访问过的顶点的路．再回溯到 h，形成路 h,i．依次回溯到 g,f,e,d，形成路 d,c,a．再回溯到 c，形成路 c,b．这样就产生了生成树．其步骤显示在图 3-10 中．

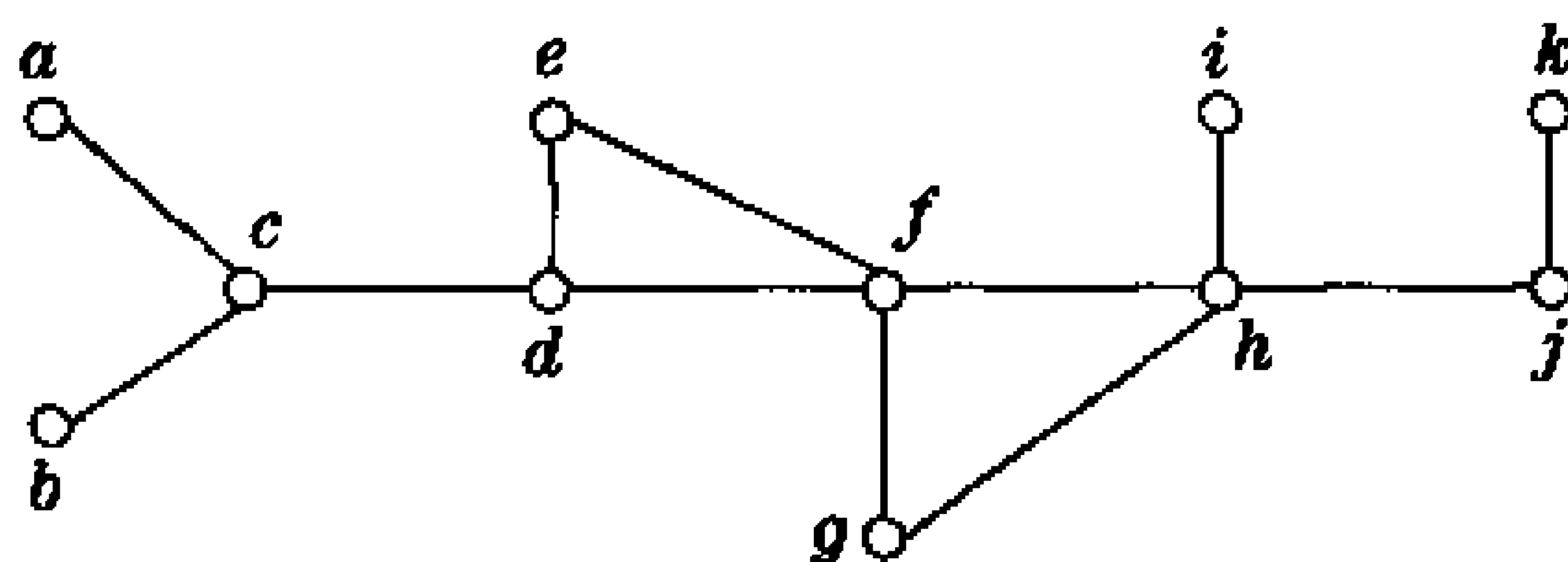


图 3-9

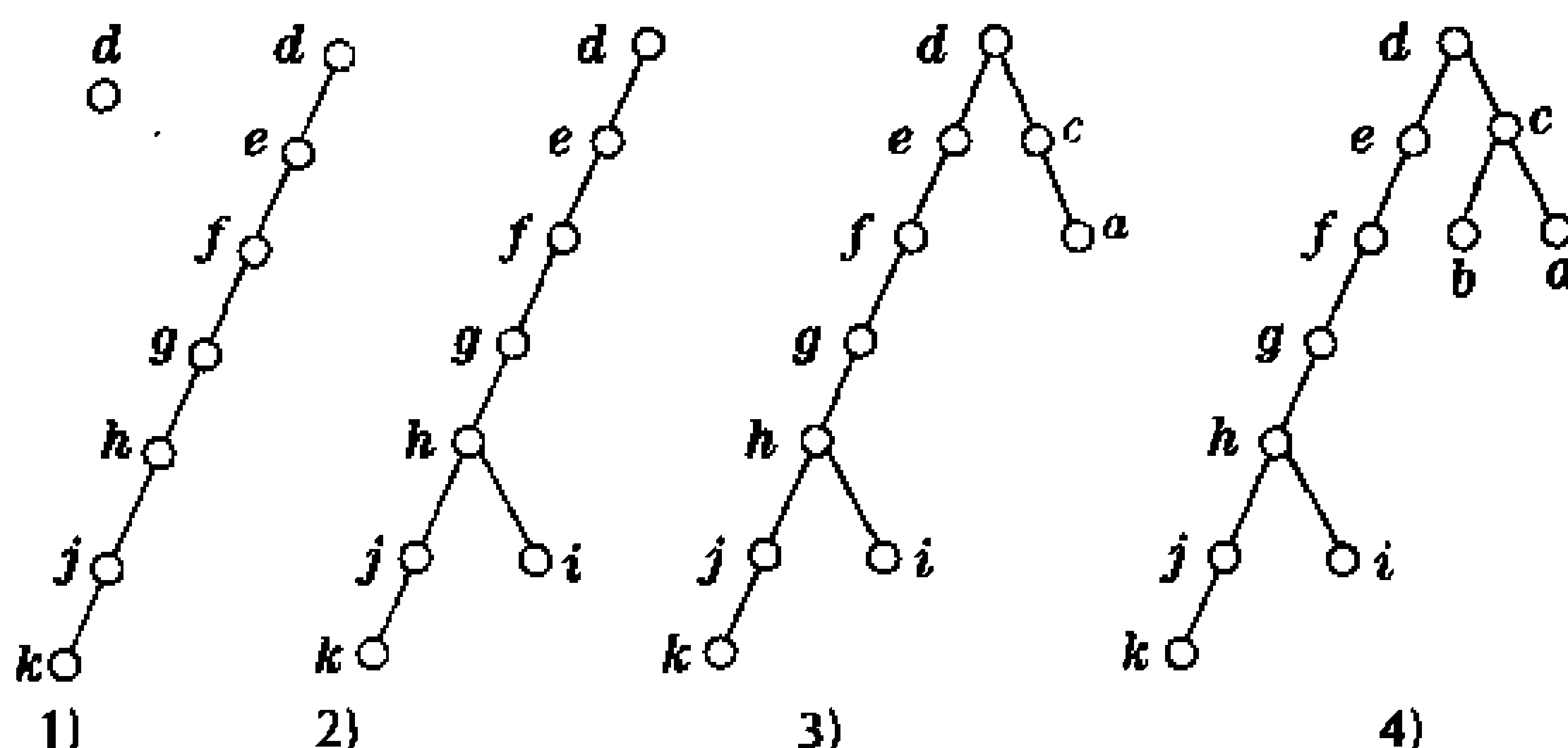


图 3-10

深度优先搜索的步骤：

设开始搜索的顶点标号为 1，用 $E(T)$ 表示所求图 G 的生成树 T 的边集，以 k 为其序号；v 为正在检查的顶点，w 为待检查的顶点，$N(i)$ 为给第 i 个顶点的标号.

第一步　$v:=1, k:=1, N(1):=1$

第二步　寻找没有检验的关联边.

(1)取与 v 关联的第一条没有检验的边，设为 $\{v,w\}$，经过此边到达顶点 w，规定边 $\{v,w\}$ 的方向从 v 到 w. 转第三步.

(2)如果没有这样的边，即与顶点 v 关联的每一边都已检验过，转第四步.

第三步　(1)若 w 是未被访问过的顶点，即 $N(w)$ 尚未确定，则把边 $\{v,w\}$ 送入边集 $E(T)$，令 $v:=w, k:=k+1, N(w):=$

k,转第二步.

(2)若 w 是已被访问过的顶点,即 $N(w)\neq 0$,回到顶点 v,转第二步.

第四步　确定集合 T 中指向顶点 v 的边$\{u,v\}$.

(1)找到该边返回到顶点 u,令 $v:=u$,转第二步.

(2)没有边$\{u,v\}$,则停止.

也可以通过使用**广度优先搜索(breadth first search)**来产生连通简单图的生成树.广度优先搜索的基本思想:从图的顶点中任意地选择一个根,然后添加与这个顶点相关联的所有边,在这个阶段添加的新顶点成为生成树里 1 层上的顶点,任意地排序它们.下一步,按顺序访问 1 层上的每一个顶点,只要不产生回路,就添加与这个顶点相关联的每条边,这样就产生了树里 2 层上的顶点.遵循同样的原则继续下去,经有限步骤后(因为图中只有有限条边)就产生了生成树.

例 2　用广度优先搜找出图 3-9 所示图 G 的生成树.

解　选择顶点 f 作为根,添加与 f 关联的所有边,所以 d,e,g,h 为 1 层顶点.再添加与 d 关联的边$\{c,d\}$,与 h 关联的边$\{i,h\}$和$\{j,h\}$,因此顶点 c,i,j 为 2 层顶点.最后添加与 c 关联的边$\{a,c\}$和$\{b,c\}$,与 j 关联的边$\{k,j\}$,因此顶点 a,b,k 为 3 层顶点.其步骤如图 3-11 所示.

最后,我们通过一个例题来强调一下回溯的用途.

例 3　n 皇后的问题:要求在 $n\times n$ 格的国际象棋盘上摆放 n 个皇后,使其不能互相攻击.即任意两个皇后都不处于同一行、同一列或同一斜线上.问有多少种摆法.

解　为解决这个问题,我们把棋盘的横坐标定为 i,纵坐标定为 j,i 与 j 的值都是从 1 到 n.为此,必须在 $n\times n$ 棋盘上找出 n 个位置,使得这些位置中没有两个是在同一行上、同一列上或在同一斜线上(斜线是由对某个 m 来说满足 $i+j=m$ 或 $i-j=m$ 的

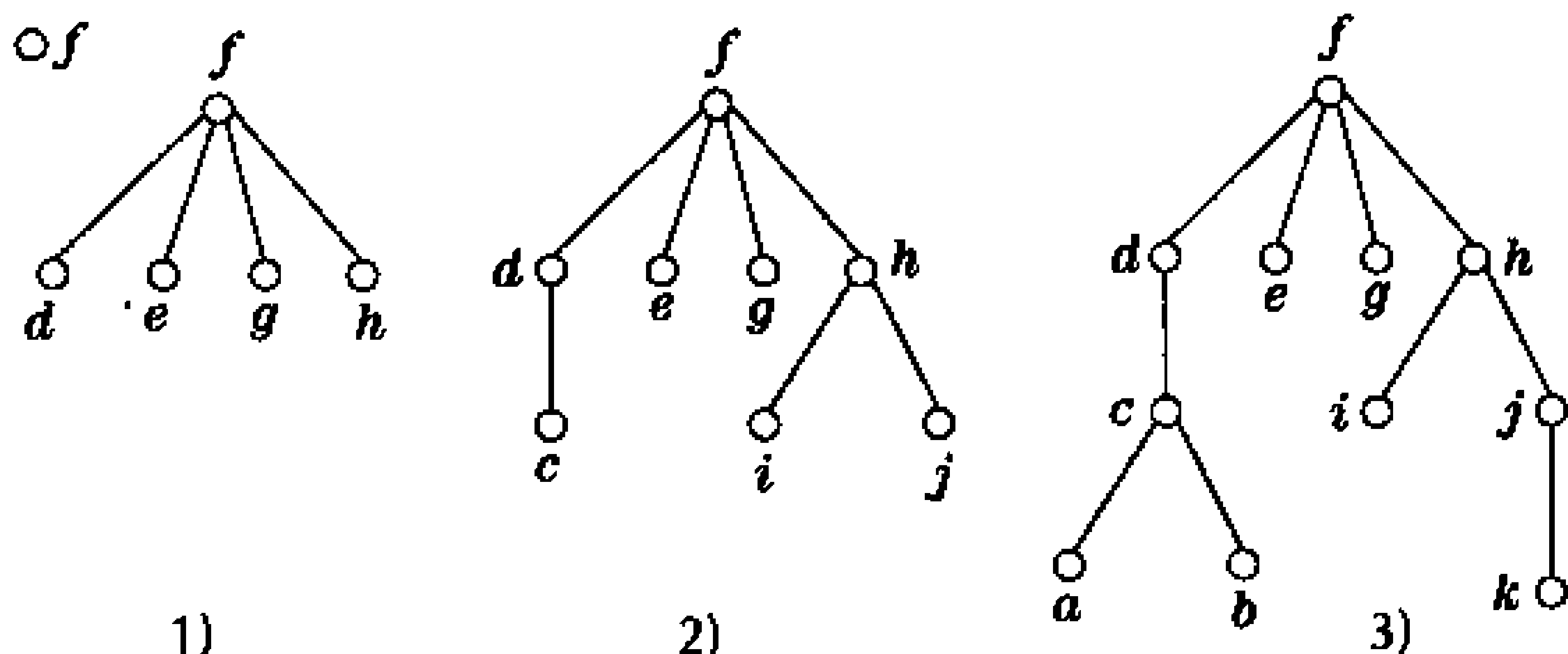


图 3-11

所有位置(i,j)组成的). 将用回溯来解决 n 皇后问题. 从空棋盘开始, 在 $k+1$ 阶段上, 尝试在棋盘上第 $k+1$ 列里放置第 $k+1$ 个皇后, 其中在前 k 列里已经有了第 k 个皇后; 从第一行开始, 检查第 $k+1$ 列格子, 寻找放置这个皇后的位置, 使它与已经在棋盘上的皇后不在同一行或同一斜线上(已经知道它不在同一列里); 若在第 $k+1$ 列里找不到放置皇后的位置, 则回溯到第 k 列, 释放第 k 个皇后原来的位置, 在该列里下一个允许行里放置皇后, 如果这样的行存在的话; 若没有这样的行存在, 则继续回溯.

在第一列位置(1,1)放置皇后 1, 然后在第二列位置(3,2)放置皇后 2; 由于在第三列找不到放置皇后 3 的位置, 于是回溯第二列, 释放皇后 2 原有位置(3,2), 在该列的下一个允许行找到可行位置(4,2)放置皇后 2; 在第三列选可行位置(2,3)放皇后 3; 由于在第四列找不到可行位置, 于是回溯第三列, 释放皇后 3 的原位置(2,3), 而在该列的下一个允许行找不到可行位置, 再回溯第二列, 释放皇后 2 的原位置(4,2), 同样由于没有下一个允许行, 继续回溯到第一列, 释放皇后 1 的原位置(1,1); 在下一个允许行找到可行位置(1,2)放置皇后 1; 依次在第二列找到位置(4,2)放皇后 2; 在第三列找到(1,3)放皇后 3; 在第四列找到(3.4)放皇后 4. 这样就得出图 3-12 所示的解.

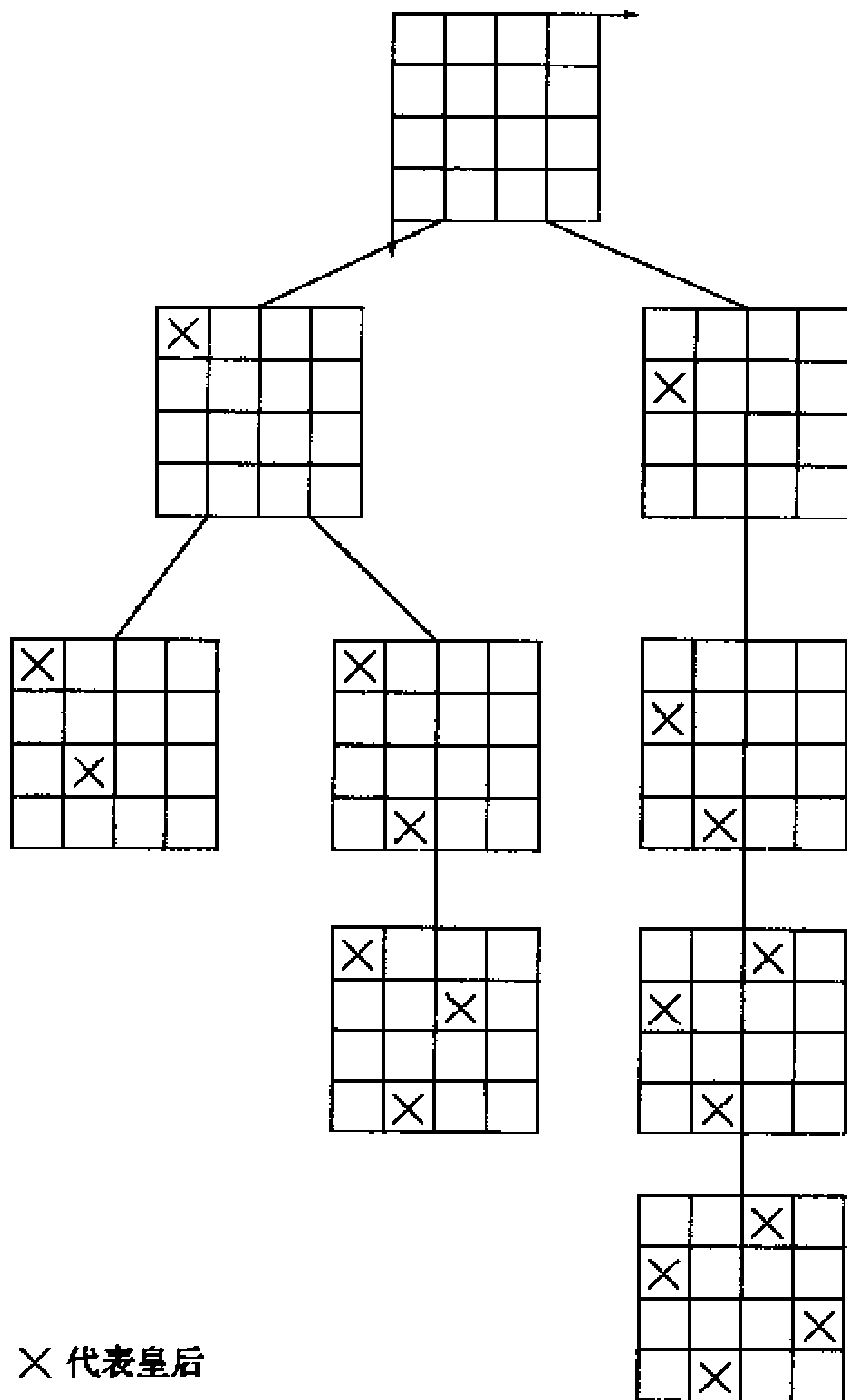


图 3-12

第四节　最小支撑树

定义 1　连通加权图里权和最小的支撑树称为**最小支撑树**(**minimum cost spanning tree**).

在加权连通图中,寻求最小支撑树的问题有其实际背景,例如在某一国家或地区,需建造一铁路网把一些城市联结起来,需要总长度最短或造价最低.

下面介绍构造最小支撑树的两个著名算法.这两个算法都是通过添加还没有使用过的具有规定性质且权最小的边来进行的.且这两个算法都是**贪心算法**的例子.贪心算法是在每个步骤上都做最优选择的算法,该原理可概括为"局部最优化".但要注意,在算法的每个步骤上都最优化,并不保证产生全局最优解.不过,本节给出的两个算法都是产生最优解的贪心算法.

我们要介绍的第一个算法是罗伯特·普林(Robert Clay Prim, 1921～)在 1957 年给出的,这个算法的基本思想:首先选择带最小权的边,把它放进支撑树里.相继向树里添加带最小权的边,这些边与已在树里的顶点相关联,并且不与已在树里的边形成圈.当已经添加了 $n-1$ 条边为止(n 为结点数).下面是其伪代码描述.

算法 1　普林算法

procedure prim(G:带 n 个顶点的连通无向图)

T:=权最小的边

for i:=1 *to* $n-2$

begin

e:=与 T 里顶点相关联的权最小的边,并且若添加到 T 里则不形成圈

T:=添加 e 之后的 T

end{T 是 G 的最小支撑树}

例 1 用普林算法求图 3-13(a)所示图的最小支撑树.

解 用普林算法所构造的最小支撑树显示在图 3-13(b)里.相继选择的边也做了显示.

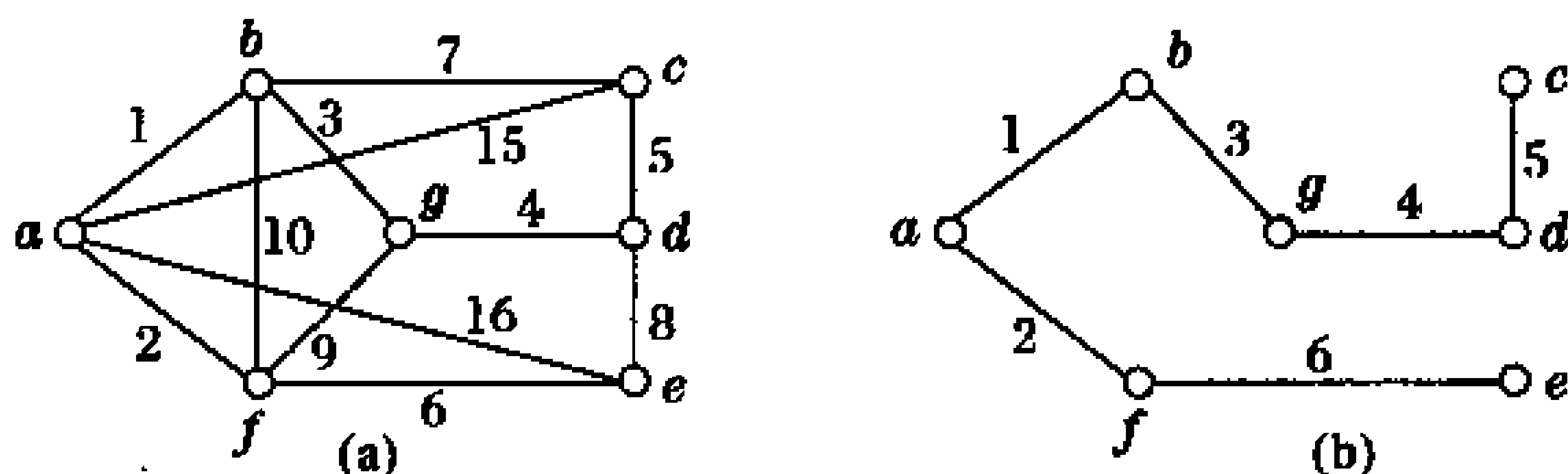


图 3-13

选择	1	2	3	4	5	6
边	$\{a,b\}$	$\{a,f\}$	$\{b,g\}$	$\{g,d\}$	$\{d,c\}$	$\{f,e\}$
权	1	2	3	4	5	6

定理 1 普林算法是正确的,即在算法结束时,得到一棵最小支撑树.

证明 设 G 是一个连通加权图.假定普林算法相继地选择的边是 $e_1,e_2,\cdots,e_{n-1}$.设 T_k 是算法 for 循环经过 k 次迭代之后所生成的树,即 T_k 的边集依次为 $e_1,e_2,\cdots,e_k$.

假定 T_k 包含在G 的最小支撑树 T 中.令 V 是 T_k 的结点集.则算法选择一条具有最小权的边$\{a,b\}$,其中 $a\in V,b\notin V$,把它加入到 T_k 中得到 T_{k+1}.如果$\{a,b\}$在 T 中,则 T_{k+1}包含在最小支撑树 T 中;如果$\{a,b\}$不在 T 中,则 $T+\{a,b\}$包含一个圈 S.在 S 中选一条不同于$\{a,b\}$的边$\{x,y\}$,其中 $x\in V,y\notin V$,则边$\{x,y\}$的权$\geqslant$边$\{a,b\}$的权.于是 $T'=T+\{a,b\}-\{x,y\}$有小于等于 T 的权.因为 T'是支撑树,因而 T'是一个最小支撑树.而 T_{k+1}包含在 T'中,从而对所有 $k=1,2,\cdots,n-1$, T_k 都包含在G 的最小支撑树里.

下面再给出普林算法的另一种形式.

设 G 是具有结点 $1,2,\cdots,n$ 的连通加权图.在每一步,一些结点具有临时标号,而另一些结点具有固定标号,结点 i 的标号表示为 L_i.若结点 i 与 j 之间有一边,则 $w(i,j)$ 等于边$\{i,j\}$的权值;若结点 i 与 j 之间没有边,则 $w(i,j)=\infty$.在图 G 中,任取一个结点为初始结点 S.

算法 2 普林算法的另一种形式

```
procedure prim-alternate{G:加权连通图}
  for
    j:=1 to to n do
  begin
    L_j:=w(s,j){临时标号}
    B(j):=s
  end
    L_s:=0
    设置 L_s 为固定的
while 遗留临时标号 do
  begin
    选择最小临时标号 L_i
    设置 L_i 为固定的
    边{i,B(i)}加入 T
    结点 i 加入 T
    for 每个临时标号 L_k do
      if w(i,k)<L_k then
      begin
        L_k:=w(i,k)
        B(k):=i
```

```
        end
      end
    return
  end {T 为 G 的最小支撑树}
```

例 2 用算法 2 求图 3-13(a)所示图的最小支撑树.

解 取结点 a 为初始点, T 为所求的最小支撑树, 如图 3-13(b)所示.

解题步骤如下:

$L_a=\infty$ $L_b=1$ $L_c=15$ $L_d=\infty$ $L_e=16$ $L_f=2$ $L_g=\infty$

$B(a)=B(b)=B(c)=B(d)=B(e)=B(f)=a$

$L_a=0$ $L_a=*$

$L_i=L_b$ $L_b=*$

$\{b,a\}\to T$ $b\to T$

$w(b,c)=7<L_c$

$w(b,d)=\infty$

$w(b,e)=\infty$

$w(b,f)=10$

$w(b,g)=3<L_g$

$L_c=7$ $B(c)=b$ $L_g=3$ $B(g)=b$

$L_i=L_f$ $L_f=*$

$\{f,a\}\to T$ $f\to T$

$w(f,c)=\infty$

$w(f,d)=\infty$

$w(f,e)=6<L_e$

$w(f,g)=9$

$L_e=6 \quad B(e)=f$

$L_i=L_g \qquad L_g=*$

$\{g,b\}\to T \qquad g\to T$

$w(g,c)=\infty$

$w(g,d)=4<L_d$

$w(g,e)=\infty$

$L_d=4 \qquad B(d)=g$

$L_i=L_d$

$\{d,g\}\to T \qquad d\to T$

$w(d,c)=5<Lc$

$w(d,e)=8$

$L_c=5 \qquad B(c)=d$

$L_i=L_c \qquad Lc=*$

$\{c,d\}\to T \qquad c\to T$

$w(c,e)=\infty$

$L_i=L_e$

$\{e,f\}\to T \qquad e\to T$

我们要介绍的第二个算法是约瑟夫·克鲁斯卡尔(Joseph Bernard Kruskal,1928～)在 1956 年发现的,其基本思想为:选择图中权最小的一条边,相继添加不与已经选择的边形成圈的权最小的边,挑选 $n-1$ 条边为止(其中 n 为结点的个数).该算法的伪代码如下.

算法 3　克鲁斯卡尔算法

```
procedure Kruskal(G:n 个顶点的连通加权无向图)
  T:=空图.
for i:=1 to n-1
begin
  e:=当添加到 T 里时不形成圈的G 里权最小的边
  T:=添加 e 之后的 T
end{T 是G 的最小支撑树}
```

例 3　用克鲁斯卡尔算法求图 3-13(a)所示图的最小生成树.

解　所构造的最小支撑树也如图 3-13(b)所示,且相继选择的边显示如下:

选择	1	2	3	4	5	6
边	$\{a,b\}$	$\{a,f\}$	$\{b,g\}$	$\{d,g\}$	$\{c,d\}$	$\{e,f\}$
权	1	2	3	4	5	6

定理 2　由 Kruskal 算法构作的任何生成树 $T=G[e_1,e_2,\cdots,e_{n-1}]$都是 G 的最小生成树,其中 n 为 G 的结点数.

证明　设 T 不是 G 的最小生成树.对 G 的任何异于 T 的生成树 T^*,定义 T^* 的函数 $f(T^*)$如下:

$$f(T^*)=\min\{i \mid e_i \text{ 不在 } T^* \text{ 中}\}$$

选取 G 的一棵最小生成树 T.使 $f(T_0)$最大.设 $f(T_0)=k$,则 $e_1,e_2,\cdots,e_{k-1}$同时在 T_0 中和 T 中,但 e_k 不在 T_0 中.因此 T_0+e_k 包含唯一的一个回路,记为 C.则 C 中至少有一条边设为 e'_k 不在 T 中.取

$$T'=(T_0+e_k)-e'_k$$

则 T'是含有 $n-1$ 条边的连通图,因此 T'也是 G 的生成树,易得

$$w(T')=w(T_0)+w(e_k)-w(e'_k) \quad (1)$$

由于 Kruskal 算法中选取的边 e_k，是使 $G[e_1,e_2,\cdots,e_k]$为无回路图的权最小的边，而 $G[e_1,e_2,\cdots,e_{k-1},e'_k]$是 T_0 的子图，它也是无回路图，于是可得

$$w(e'_k)\geqslant w(e_k) \quad (2)$$

结合式(1)和式(2)有 $w(T')\leqslant w(T_0)$

所以 T'也是G 的一棵最小支撑树. 然而，由于$\{e_1,e_2,\cdots,e_k\}\subseteq E(T')$，我们有

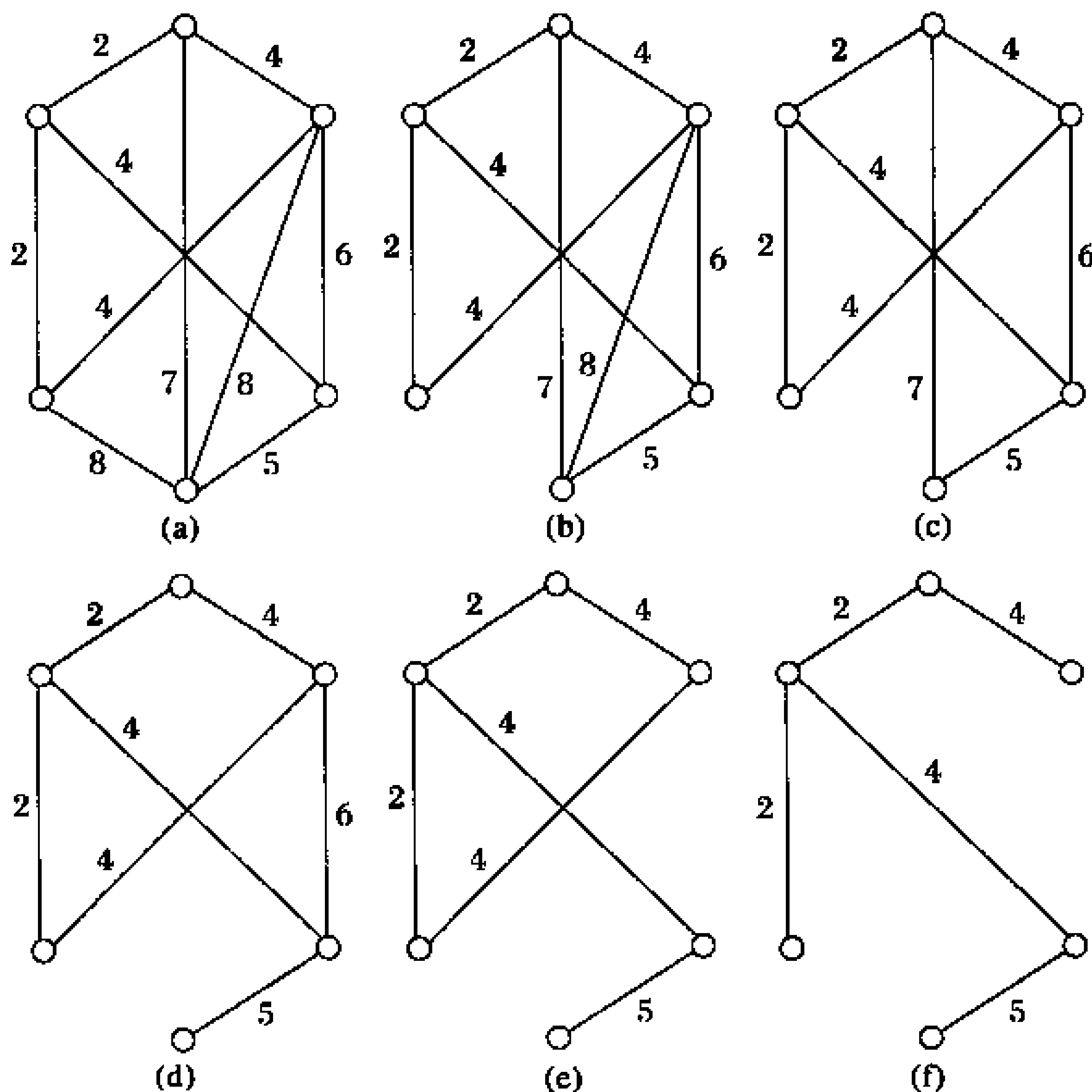


图 3-14

$$f(T')>k=f(T_0)$$

与 T_0 的选取相矛盾. 因此 T 是 G 的最小支撑树. 证毕

下面我们再介绍求最小生成树的**破圈法**,它是管梅谷教授给出的.

设 G 是连通加权简单图,若 G 不是树,则 G 中必有回路,我们删去 G 中含于某回路内权最大的一条边,所得图记为 G_1, G_1 是 G 的连通生成子图. 下一步,若 G_1 不是树,又从 G_1 的某回路内删去权最大的一条边,如此下去,最后不能按上述方式删边时,得到的图 T 便是 G 的一棵生成树. 不难证明, T 是 G 的最小生成树.

已知连通加权简单图 G 如图 3-14(a)所示. 图 3-14($b-f$)显示了用破圈法逐次去边得到的 G_k. 其中图 3-14(f)是 G 的最小生成树.

第五节 前缀码

定义 1 设 T 是有 t 片叶子的二叉树. 其中 t 片叶子分别带有权 $w_1, w_2, \cdots, w_t$,称 T 为**加权二叉树**. 称 $W(T)=\sum_{i=1}^{t} w_i l_i$ 为二叉树 T 的权,其中 l_i 为带权 w_i 的树叶 v_i 的层数. 在所有的带权 $w_1, w_2, \cdots, w_t$ 的二叉树中,带权最小的二叉树称为**最优二叉树**.

最优二叉树问题源于计算机科学、生产管理等领域. 1952 年哈夫曼(D. A. Huffman)给出了求最优二叉树的算法. 该算法的核心思想是从带权 $w_1+w_2, w_3, \cdots, w_t$ 的最优二叉树可得到带权 $w_1, w_2, \cdots, w_t$ 的最优二叉树. 其步骤如下:

· 给定实数 $w_1, w_2, \cdots, w_t$ 且 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$.

(1)连接 w_1, w_2 为权的两片树叶,得一分枝点,其权为 $w_1 + w_2$.

(2)在 $w_1 + w_2, w_3, \cdots, w_t$ 中选出两个最小的权,连接它们对应的顶点(不一定都是树叶)得分枝点及所带的权.

(3)重复(2),直到形成 $t-1$ 个分枝点,t 片树叶为止.

引理 1 存在一棵带权 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$ 的最优二叉树 T,使在 T 中,一定能使带权 w_1, w_2 的顶点为兄弟,且它们的层数相同,均为树高.

证明 设 T 是带权 $w_1, w_2, \cdots, w_t$ 的最优二叉树.在 T_1 中,设 v 是层数最大的分枝点之一,它的两个儿子 v_a 和 v_b 都是树叶,分别带权 w_a 和 w_b,而不是 w_1 和 w_2.并且它们的层数分别为 $l_a = l(v_a)$ 和 $l_b = l(v_b)$,$l_a = l_b$,且均为 T_1 的高.现在把树叶 v_a 和 v_b 所带权 w_a 和 w_b 分别与 w_1 和 w_2 交换,得到一棵新的带权二叉树 T_2.下面证明 T_2 也是带权 $w_1, w_2, \cdots, w_t$ 的最优二叉树,且带权 w_1 和 w_2 的树叶是兄弟.因为 w_1, w_2 是最小的两个权.又

$$W(T_1) = w_1 l(v_1) + w_2 l(v_2) + \cdots + w_a l(v_a) + w_b l(v_b) + \cdots$$

$$W(T_2) = w_a l(v_1) + w_b l(v_2) + \cdots + w_1 l(v_a) + w_2 l(v_b) + \cdots$$

于是 $W(T_1) - W(T_2) = (w_a - w_1)(l(v_a) - l(v_1)) + (w_b - w_2)(l(v_b) - l(v_2)) \geqslant 0$

所以 $W(T_1) \geqslant W(T_2)$,又因 T_1 是带权 $w_1, w_2, \cdots, w_t$ 的最优二叉树,因此 $W(T_2) = W(T_1)$.从而可知 T_2 是带权 $w_1, w_2, \cdots, w_t$ 的最优二叉树,且带权 w_1, w_2 的树叶是兄弟,其层数为树高.

定理 1 设 T 为带权 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$ 的最优二叉树,若将以带权 w_1 和 w_2 的树叶为儿子的分枝点改为带权 $w_1 + w_2$ 的树

叶,得到一棵新树 T',则 T'也是最优二叉树.

证明 由题设,有 $W(T)=W(T')+w_1+w_2$.

若 T'不是最优二叉树,则必有另一棵带权 $w_1+w_2,w_3,\cdots,w_t$ 的最优二叉树T''.对 T''中带权w_1+w_2 的树叶 $V_{w_1+w_2}$生成两个儿子,得到新树 $\bar{T}$,则 $W(\bar{T})=W(T'')+w_1+w_2$.因为 T''是带权$w_1+w_2,w_3,\cdots,w_t$ 的最优树,故 $W(T'')\leqslant W(T')$.若 $W(T'')<W(T')$,则 $W(\bar{T})<W(T)$,这与 T 是带权$w_1,w_2,\cdots,w_t$ 的最优二叉树矛盾,因此 $W(T'')=W(T')$,即 T'是带权$w_1+w_2,w_3,\cdots,w_t$ 的最优二叉树.证毕

由上述定理,要画一棵带有 t 个权的最优二叉树,可简化为画一棵带有 $t-1$ 个权的最优二叉树,而这又可简化为画一棵带有 $t-2$ 个权的最优二叉树,依此类推.

例 1 求带权为 1,3,5,7,8,11,13 的最优二叉树.

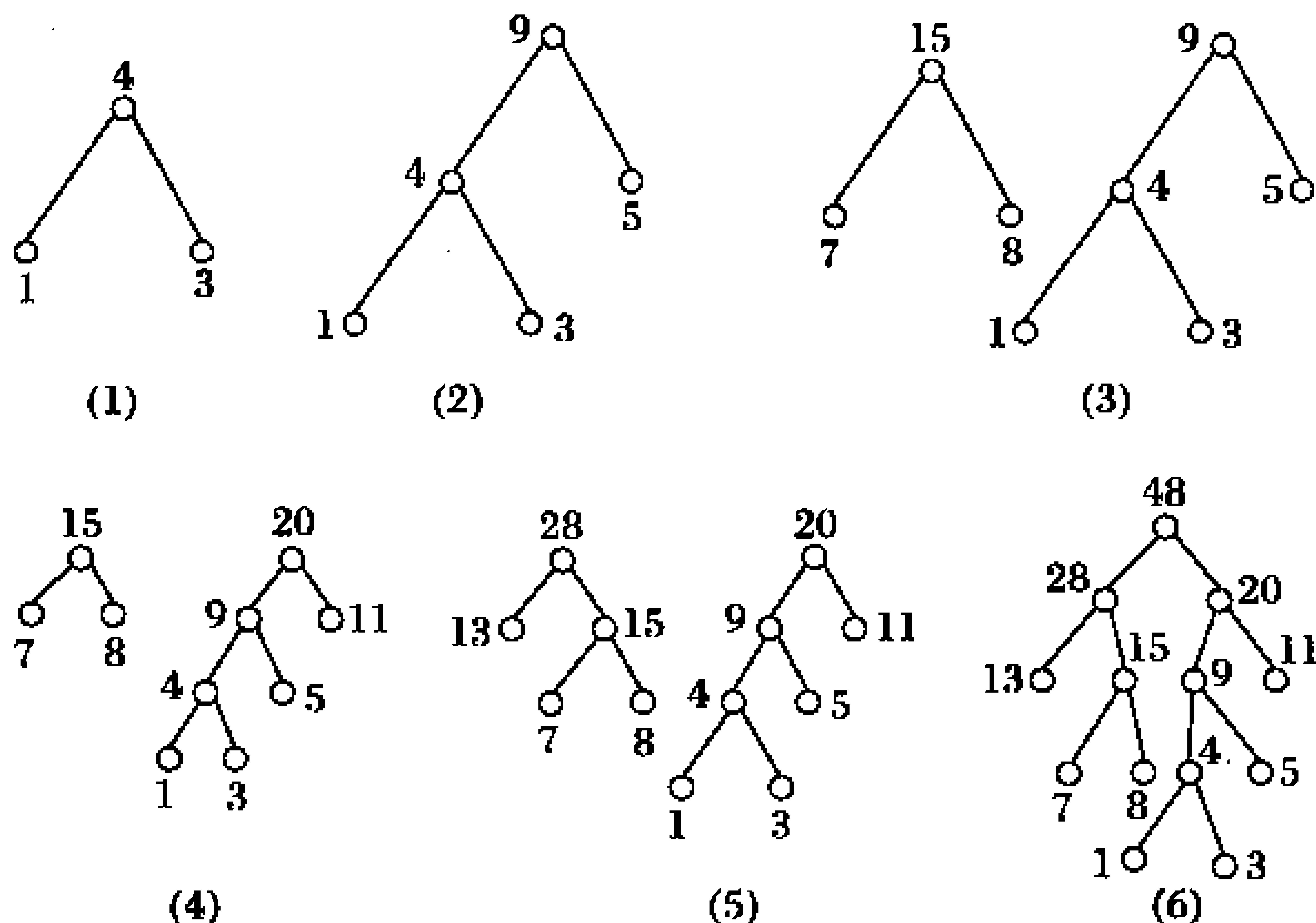


图 3-15

解　图 3-15 给出了求最优二叉树的过程，且 $W(T)=124$.

利用哈夫曼算法可以产生最优前缀码，首先介绍前缀码.

通信中，常用 0 和 1 组成的二进制序列表示 26 个英文字母. 我们知道，长度为 1 的二进制序列有 2 个，长度为 2 的二进制序列有 2^2 个，长度为 3 的二进制序列有 2^3 个，…，长度为 i 的二进制序列有 2^i 个. 因此，用长度不超过 4 的二进序列就可以表达 26 个不同英文字母. 由于字母使用的频率不同，为尽量缩小传输量，把出现频率高的字母用较短的二进制序列表示. 但在接收端，必须把收到的一长串二进制序列分割成单个字母对应的二进制序列，即正确译码. 比如，我们用 00 表示 e，用 01 表示 t，用 0001 表示 q，那么当接收端收到信息串 0001 时，就不能决定传递的信息是 et 还是 q？为此我们要引入前缀码.

定义 2　有一个序列的集合，如果在这个集合中，任何序列都不是另一个序列的前缀，则称这个集合为**前缀码**.

例如，001 是 001011 的前缀，而不是 010011 的前缀. 集合 {000,001,01,10,11} 是前缀码，而 {1,00,01,000,0001} 不是前缀码.

由一棵二叉树可以产生一个前缀码. 将一棵二叉树的每个分枝点与它左儿子之间的边记为 0，和它右儿子之间的边记为 1，把从根到每个叶子所经过边的记号序列作为叶子的记号，这些叶子标记的集合就是一个前缀码.

例如，图 3-16 所示二叉树产生的前缀码为

{00,10,011,111,0100,0101}

对 26 个英文字母，设各字母使用的频率分别为 $P_1, P_2, \cdots, P_{26}$，求出带权 $P_1, P_2, \cdots, P_{26}$ 的最优二叉树，从而解决最佳编码问题.

最后要说明的一个问题，即任何一棵有序树都可以把它改写为一棵对应的二叉树. 其一般步骤为：(1)从根结点开始，保留根结

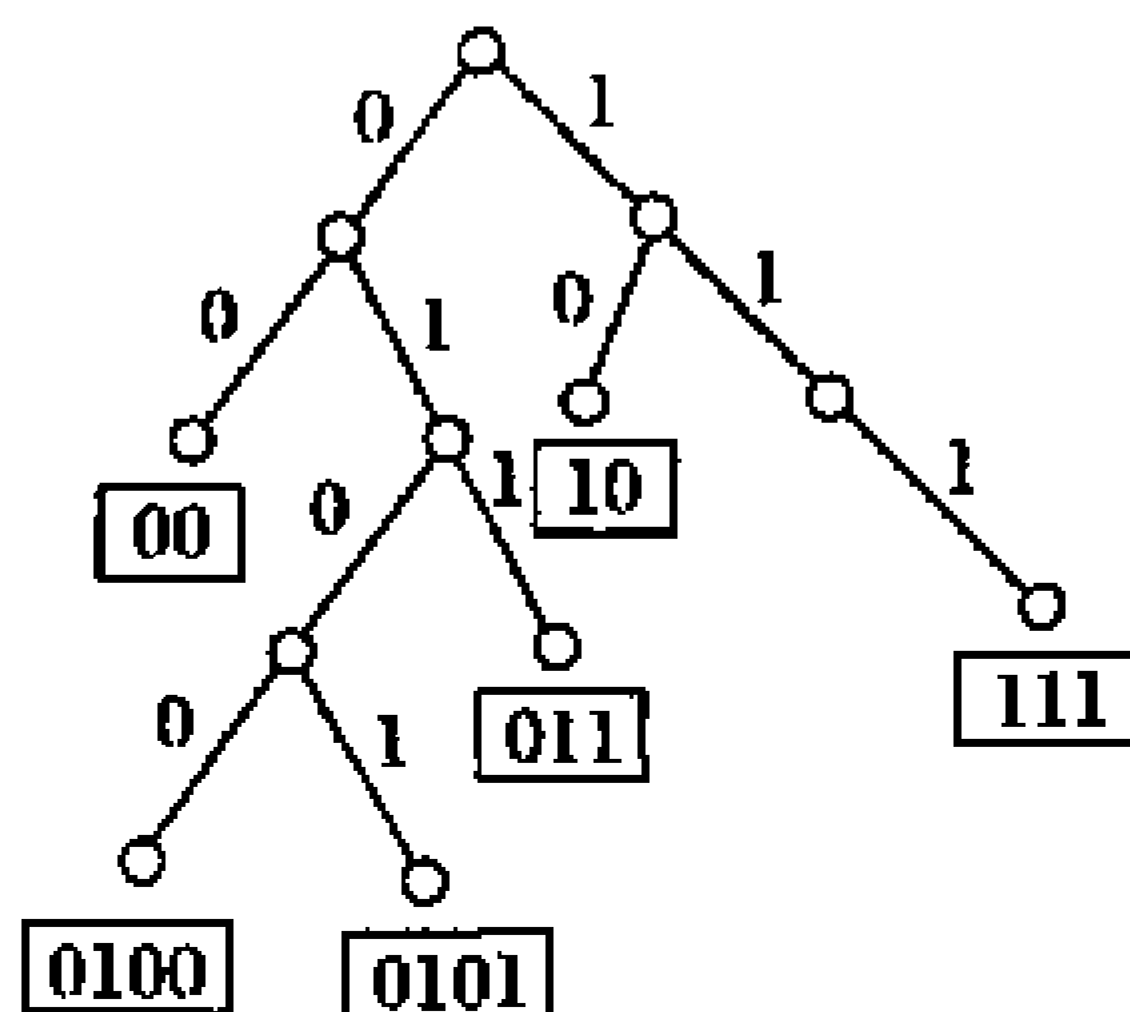


图 3-16

点同其最左边儿子的连线，删去与其它儿子的连线.(2)兄弟间从左向右加线连接.(3)用如下方法选定二叉树的左儿子和右儿子：直接处于给定结点下面的结点作为左儿子；对于同一水平线上与给定结点右邻的结点作为右儿子，依次类推.

例 2 将图 3-17(a)所示树转化为二叉树.

解 对(a)进行步骤(1)、(2)得 3-17(b)图，再按(3)得 3-17(c)图.则 3-17(c)图即为所求的二叉树.

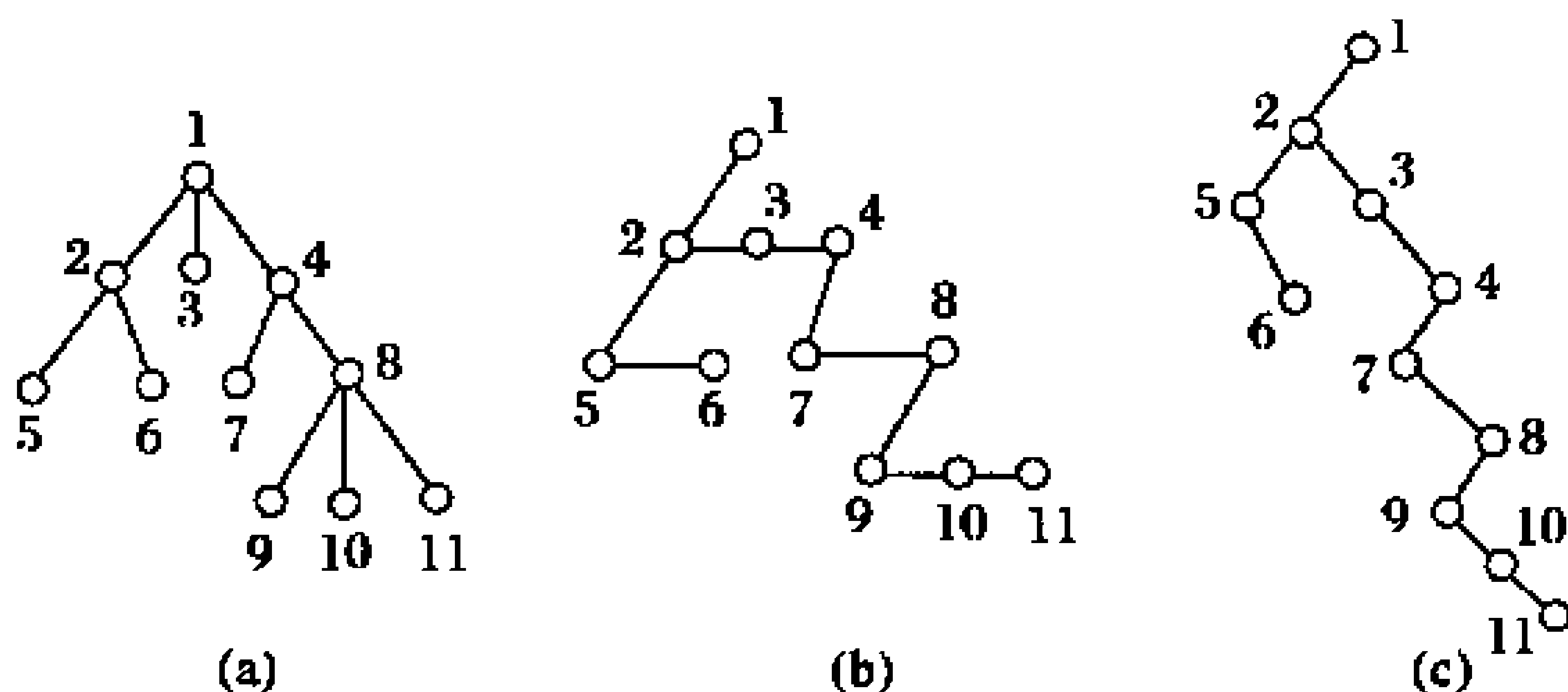


图 3-17

反过来，我们也可将图 3-17(c)还原为 3-17(a).

一个森林也可转换为二叉树.其步骤为：

(1)先把森林中每一棵树表示成一棵二叉树.

(2)除第一棵二叉树外,依次将每棵二叉树作为左边二叉树的根的右子树,直到所有的二叉树都连成一棵二叉树为止.

例 3 将图 3-18(a)所示的森林转为一棵二叉树.

解 如图 3-18(b)所示的二叉树即为所求.

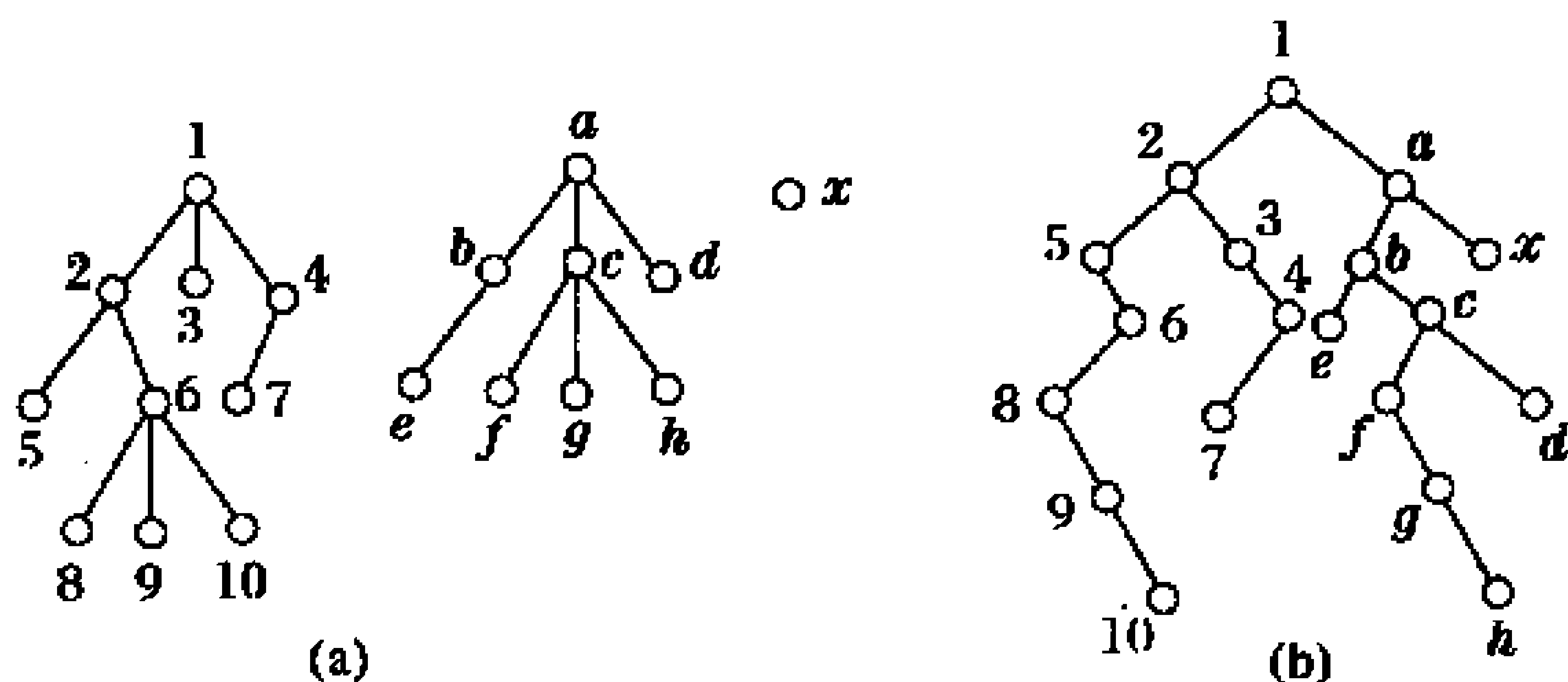


图 3-18

当然,也可以将一棵二叉树转换成森林.

第六节 二叉查找树与决策树

设有一个集合 S,S 中的元素可以被排序.例如,若 S 中元素为数字,我们可以用普通的数字序;若元素是字符串,我们采用字典序.在计算机科学中,二叉树被广泛用来存储可排列集合的元素.如果数据 $d(v)$被存于结点 v 中,数据 $d(w)$被存于结点 w 中,为保证 $d(v)$和 $d(w)$之间存在着某种顺序关系,可以选择二叉查找树.

定义 1 一棵二叉查找树是一棵二叉树,其数据与结点有关.数据被用以下方式组织起来,对树中某个结点 v 而言,其左子树中每个结点的数据都小于 v 中的数据;其右子树中每个结点的数据都大于 v 中的数据.**

下面的递归过程用来形成项的列表的二叉查找树. 从只包含一个顶点(即根)的树开始. 指定列表中第一个项作为这个根的关键字. 为了添加新的项, 首先比较它与已经在树里的顶点的关键字, 从根开始, 若这个项小于所比较顶点的关键字而且这个顶点有左儿子, 则向左移动, 或者若这个项大于所比较顶点的关键字而且这个顶点有右儿子, 则向右移动. 当这个项小于所比较顶点的关键字而且这个顶点没有左儿子时, 就插入以这个项作为关键字的一个新顶点来作为这个顶点的左儿子. 同理, 当这个项大于所比较顶点的关键字而且这个顶点没有右儿子时, 就插入以这个项作为关键字的一个新顶点来作为这个顶点的右儿子. 用下面的例子来说明这个过程.

例 1 用字母顺序建立下面这些单词的二叉查找树: oenology, phrenology, campanology, ornithology, ichthyology, limnology, alchemy, astrology.

解 可以放在如图 3-19 所示的二叉查找树中.

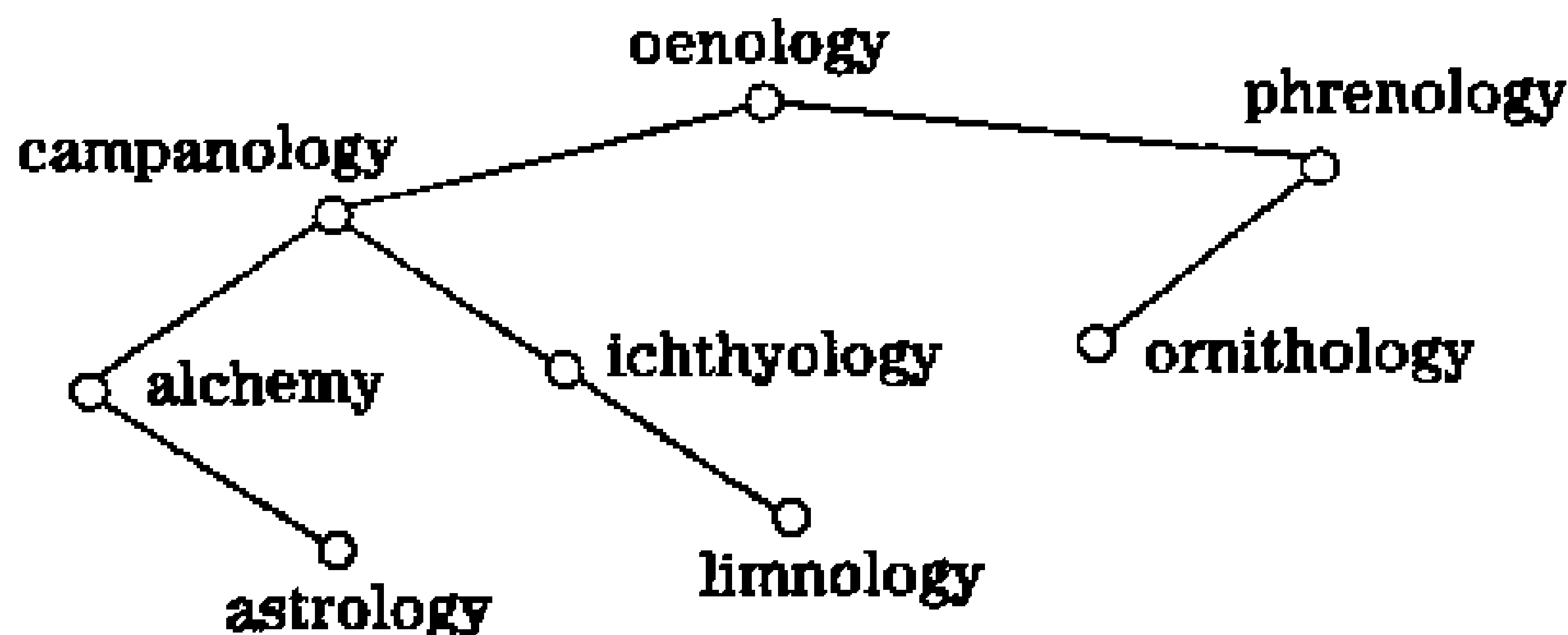


图 3-19

使用二叉查找树对于查找数据很有帮助, 也就是说, 给定数据 D, 我们很容易确定 D 是否在一棵二叉查找树中. 如果在, 还可确定其位置. 要确定数据 D 的位置, 我们要从树根开始, 比较 D 与当前结点中的数据, 若 D 与当前结点中数据一致, 我们便找到了 D, 并停止; 若 D 小于当前结点中数据, 我们就移向左孩子, 并重

复这个过程；若 D 大于当前结点中数据，我们就移向右孩子，并重复这个过程；一旦当前结点为空，我们就可以认定 D 不在树中.

下面我们来介绍决策树.每个内点都对应着一次决策，这些顶点的子树都对应着该决策的每种可能结果，这样的有序树称为**决策树**.

例 2 有 8 枚外观相同的硬币，其中有 7 枚重量相同，而剩余的一枚伪币的重量较轻.要求以比较重量的方法用一架天平去找出伪币.

解 为了便于描述这一问题的解决过程，我们用 1~8 标记硬币.每次量衡有 3 种可能：左盘低下，保持水平，右盘低下.图 3-20 给出这一解决过程的决策图.图中 $\varnothing$ 表示不会出现的结果.决策树的结点左侧标记着状态，这里表示包含有伪币的硬币集合，右侧标记测试内容.

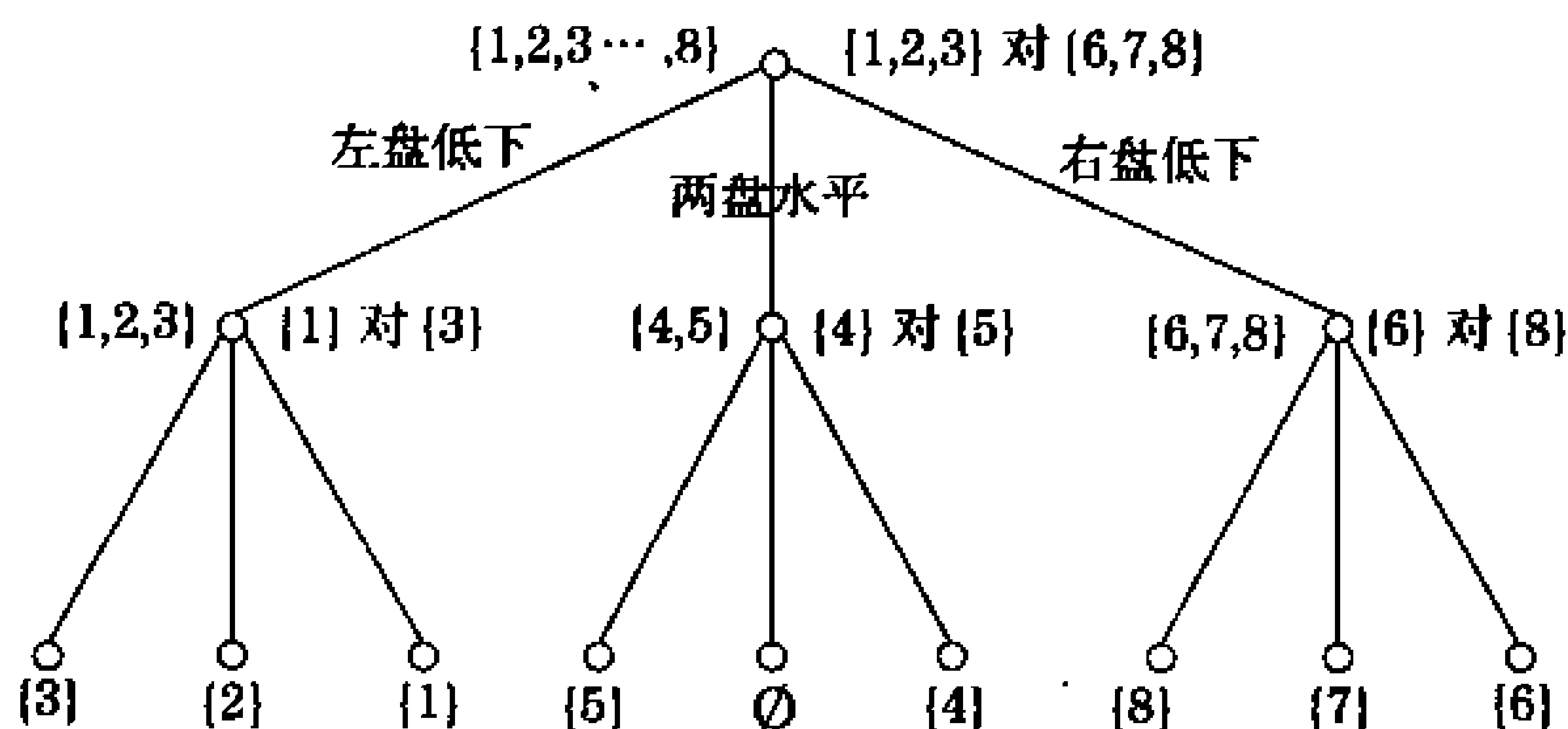


图 3-20

以上例子说明：决策树的每一内部结点对应于一个部分解；每个叶子对应于一个解.每一内部结点联结于一个获得新信息的测试.从每一结点出发的每一枝标记着不同的测试结果.一个解决过程的执行对应于通过从根到叶的一条路.一个决策树是所有可能

的解决路的集合.

例 3 有 5 枚外观相同的硬币,其中有 4 枚重量相同,而剩余的一枚伪币的重量或轻或重.要求以比较重量的方法用一架天平找出其中的伪币.

解 仍用 1~5 标识硬币.图 3-21 给出这一解决过程的决策图.

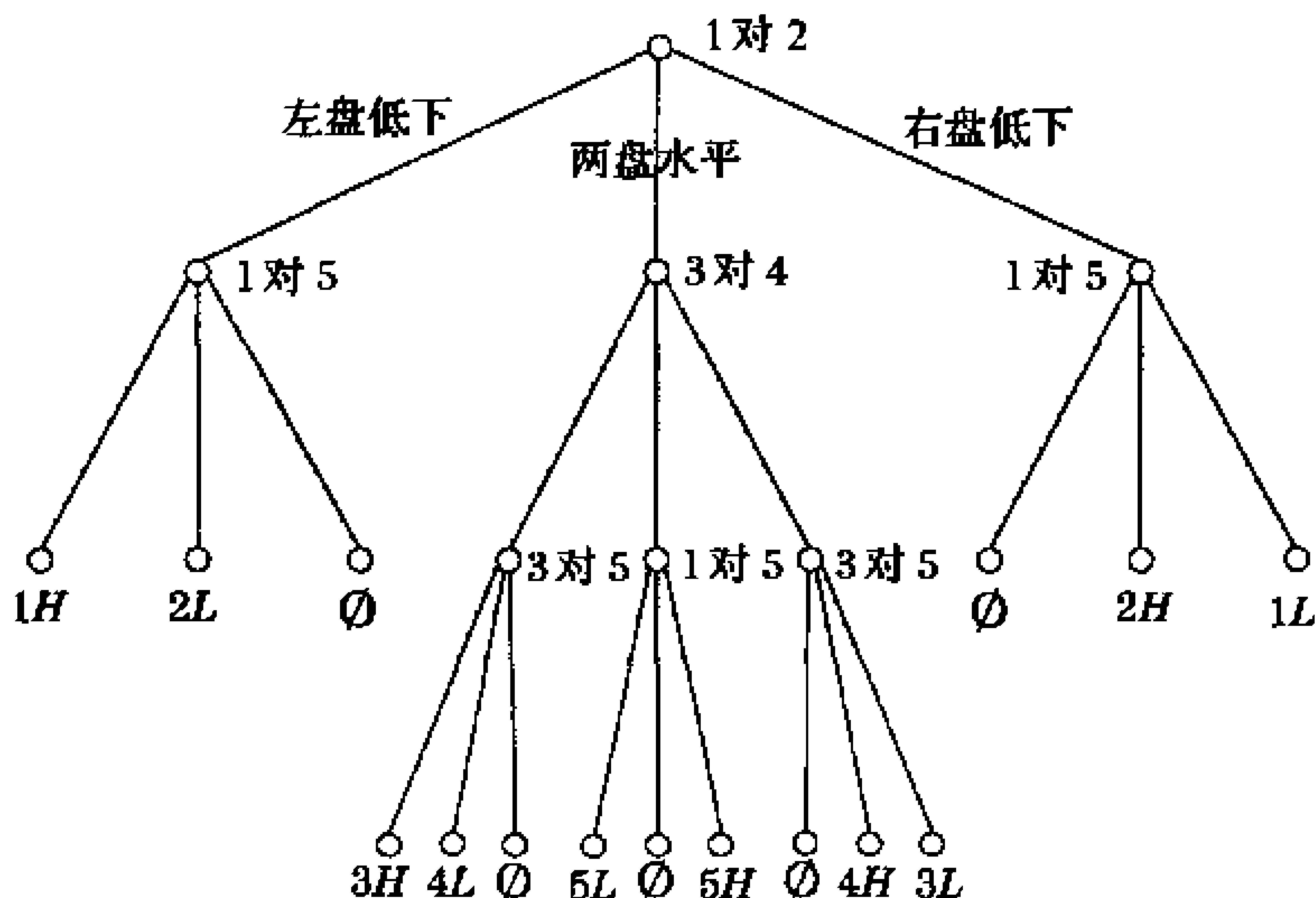


图 3-21

需要注意,一个决策树是一个算法的描述,而一个查找树是一个数据结构.例如,图 3-22(a)所示的是一棵二叉查找树,结点标记 $i(j)$,这里 i 是结点索引号,而 j 是存贮于该结点的记录的键值.对应于搜索这棵树的一棵决策树如图 3-22(b)所示,图中加方框的"无"字表示没有所找的键值.

我们还可以用决策树估计排序的最坏情况时间.排序问题可描述为:将 n 个元素 $x_1, x_2, \cdots, x_n$ 接升序或降序排列.这里将研究的排序算法都是基于二叉比较,即一次比较两个元素.每次这样

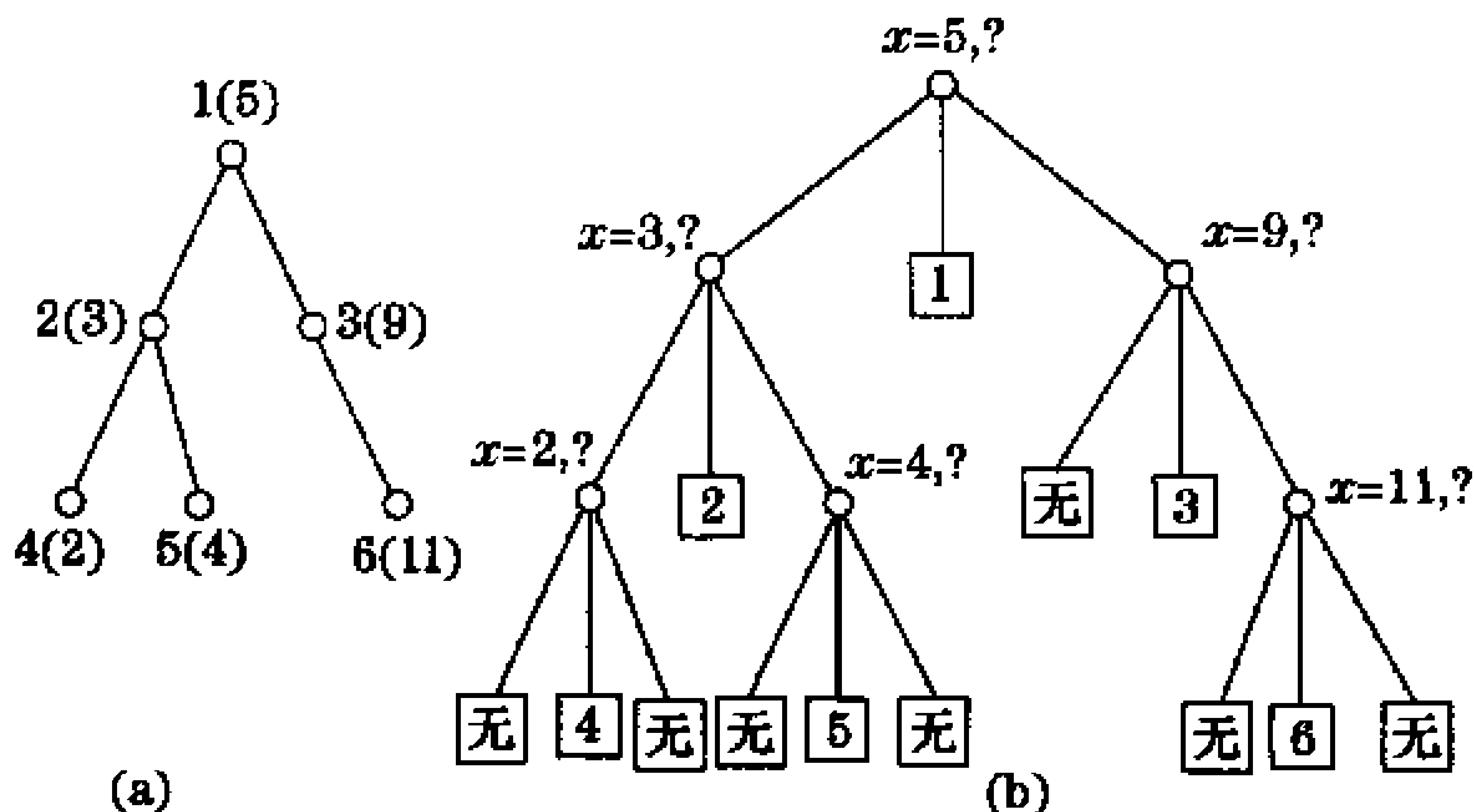


图 3-22

的比较都缩小了可能的排序集合.因此,基于二叉比较的排序算法可以表示成二叉决策树,其中每个内点表示两个元素的一次比较.每个树叶表示 n 个元素的 $n!$ 种排列中的一种.

例 4 图 3-23 所示的决策树给出了一个对 a,b,c 进行排序的算法.

对基于二叉比较的排序来说,其复杂性是通过所用的二叉比较的次数来度量的.排序有 n 个元素的列表所需要的最多比较次数就给出了这个算法的最坏情形复杂性.所用的最多比较次数等于表示这个排序过程的决策树的高度.因为带 $n!$ 个树叶的二叉树的层数至少是 $\lceil \log n! \rceil$($\lceil x \rceil$是大于或等于 x 的最小整数).

定理 1 基于二叉比较的排序算法至少需要$\lceil \log n! \rceil$次比较.

我们知道 $\log n! = O(n\log n)$.因此,若一个排序算法具有 $O(n\log n)$的时间复杂性,在时间复杂性的大 O 估计的意义下,则它就是尽可能地有效的.

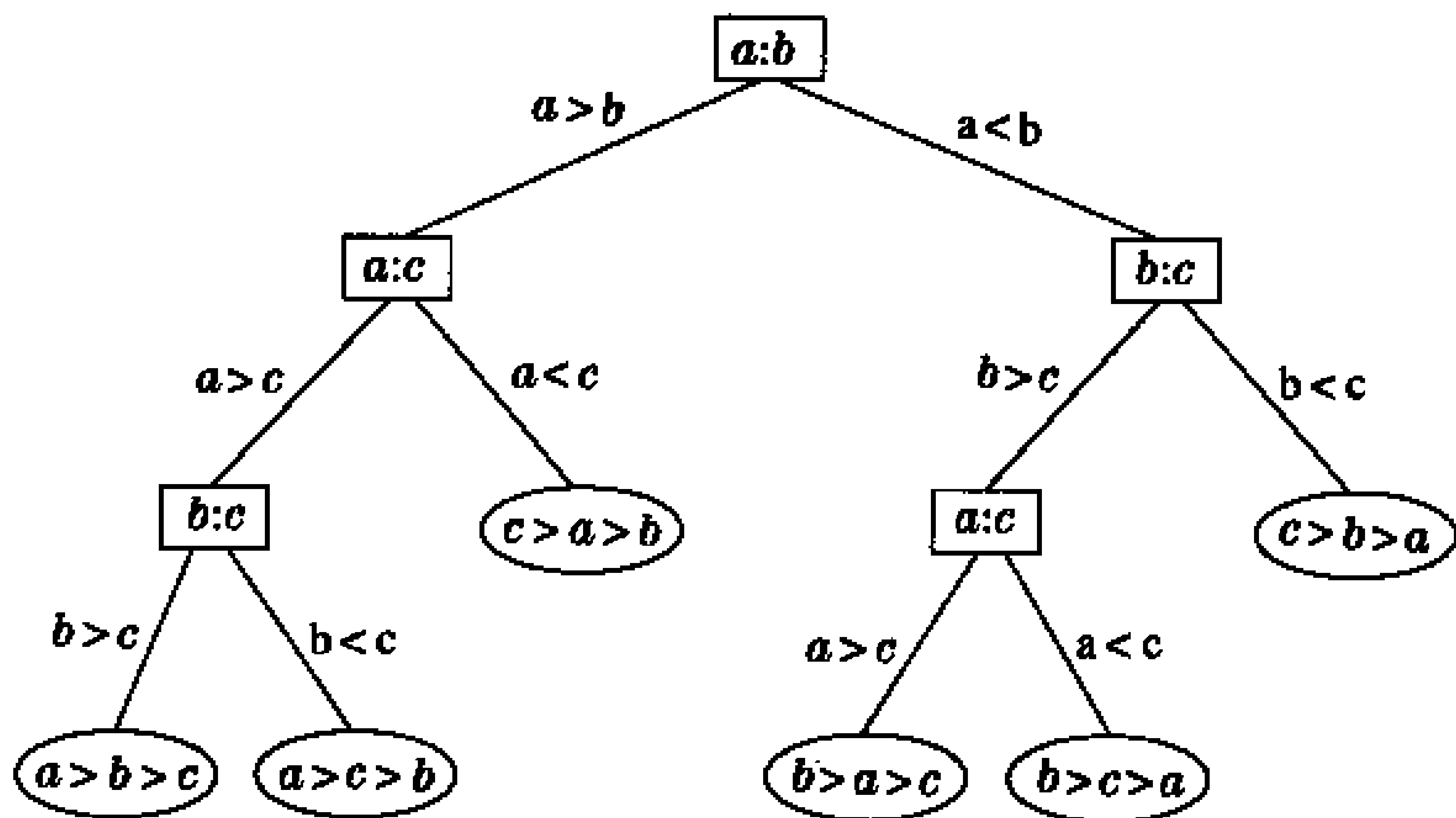


图 3-23

习　题　三

1. 设无向图 G 是由 $k(k\geqslant 2)$ 棵树构成的森林，已知 G 中有 n 个顶点，m 条边，则 $m=n-k$.

2. 设无向图 G 中有 n 个顶点 m 条边，且 $m\geqslant n$，则 G 中必有圈.

3. 已知一棵无向树 T 有三个 3 度顶点，一个 2 度顶点，其余的都是 1 度项点，试求 T 中叶子数.

4. 用图 3-24 中所示的三个星形图的并，能构造出多少棵非同构的树?

5. 画出具有七个顶点的所有非同构的树.

6. 有多少棵非同构的带有 5 个顶点的根树.（使用有向图的同构）

7. 假定 1000 个人参加围棋巡回赛.若一个选手输掉一盘就遭到淘汰，而且比赛进行到只有一位参加者还没有输过为止，则利

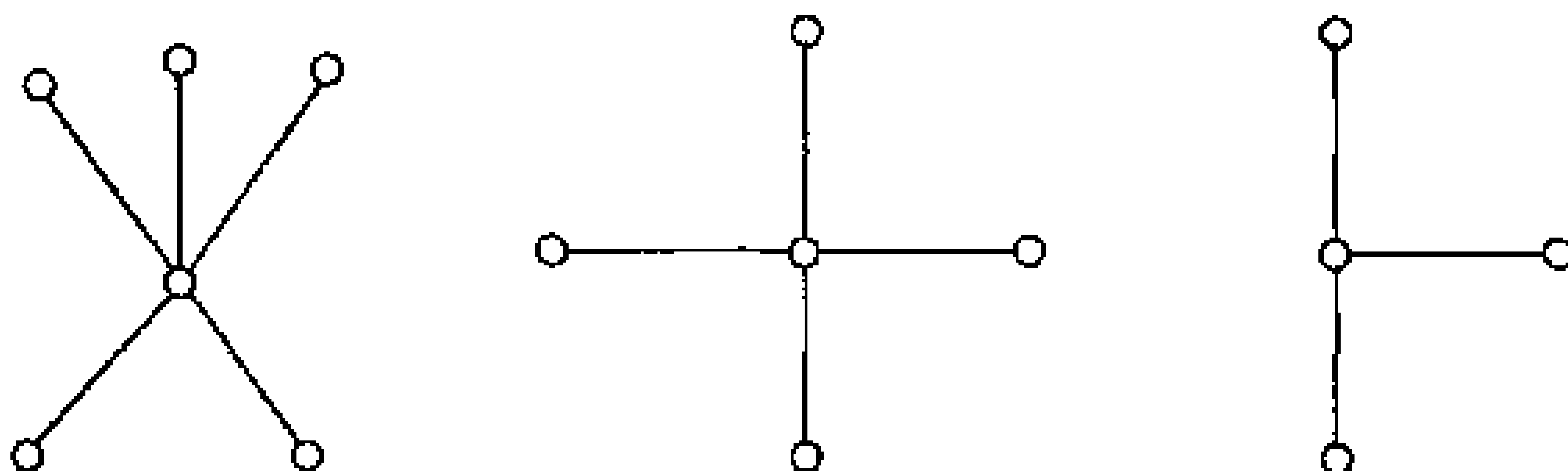


图 3-24

用这个巡回赛的根树模型，来确定为了决出冠军必须下多少盘棋？（假定没有平局）

8. 证明，在高度为 h 的 m 叉树里至多有 m^h 个树叶.（对高度使用数学归纳法）

9. **标记树**是其中每个顶点都指定了标记的树. 当在两个标记树之间存在保持顶点标记的同构时，就把这两个标记树当作同构的. 过 3 个顶点的不同构的标记树有多少棵？过 4 个顶点呢？过 n 个顶点呢？

10. 证明树是一个二分图. 哪些完全二分图 $K_{m,n}$ 是树？

11. 在图 3-25(a)所示的加权图中共有多少棵生成树，它们的权各为多少？

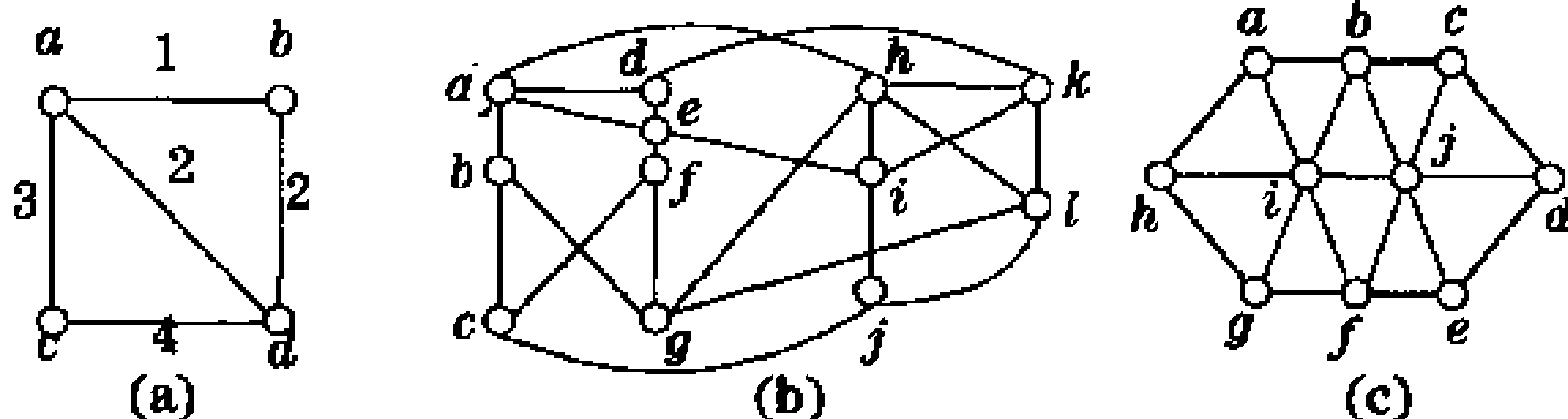
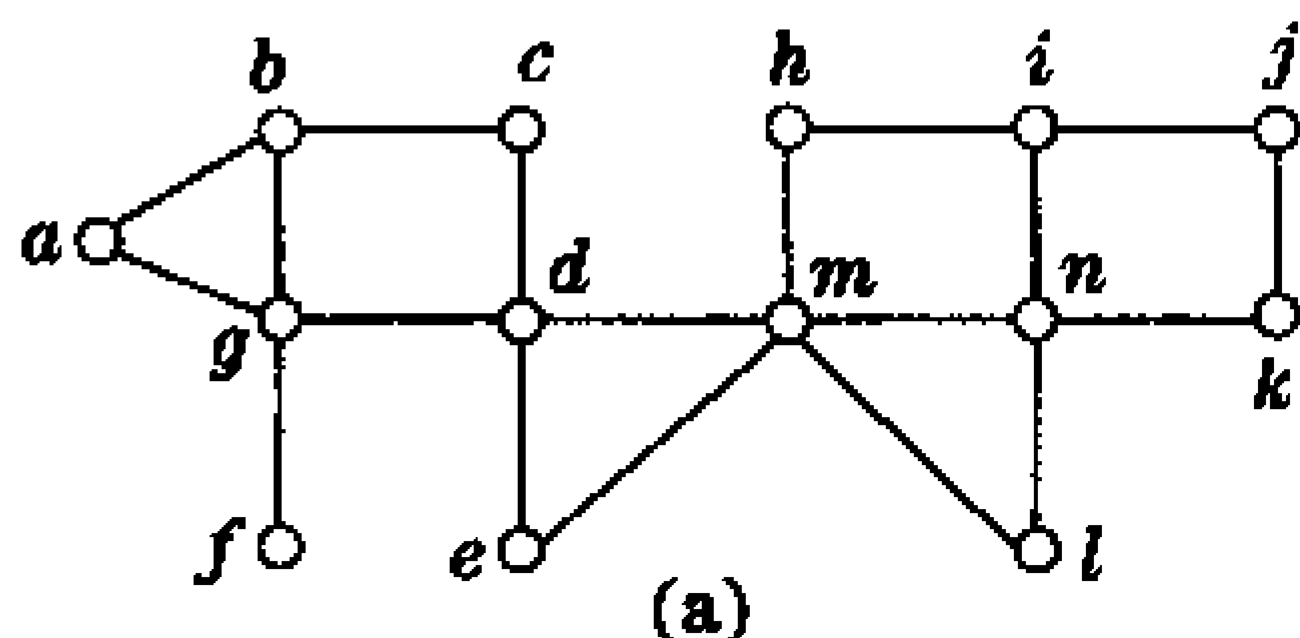


图 3-25

12. 通过删除圈里的边来分别求图 3-25(b)(c)的生成树.

13. 用深度和广度优先搜索分别构造图 3-26(a)(b)的生成树.



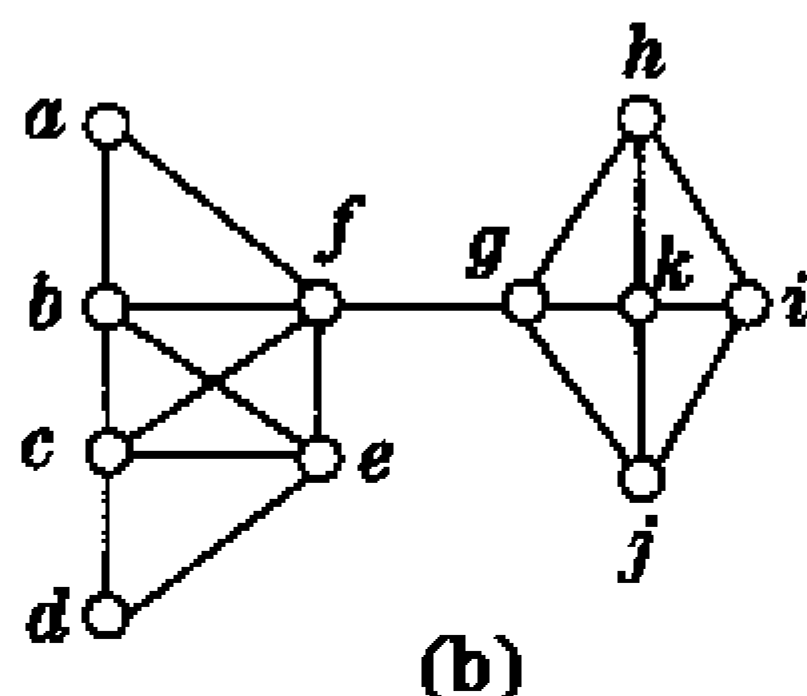


图 3-26

14. 证明,每个顶点的入度和出度都相等的弱连通有向图有根生成树.(使用欧拉回路)

15. 为 5 皇后和 6 皇后问题找到一个解.

16. 写一个广度优先搜索算法检查图是否连通.

17. 写一个深度优先搜索算法检查图是否连通.

18. 如果连通加权图 G 的权值互不相同,则 G 有唯一一棵最小生成树.

19. 如果 e 是 G 的一条边,它的权值比其他所有边的权值都要小,则 e 存在于 G 的每个最小生成树中.

20. 求有 t 个叶子的完全二叉树的最大高度.

21. 构造单词 old, programmers, never, die, they, just, lost, their, memories 的二叉查找树.

22. 构造单词 mathematics, physics, geography, zoology, meteorology, geology, psychology, chemistry 的二叉查找树.

23. 12 枚外观相同的硬币,其中有一枚比其他的或轻或重. 试用决策树描述一个算法,使得只用一个天平且最多进行三次比较就可以确定出坏币并且判断出它是轻是重.

24. 4 枚外观相同的硬币,其中有 3 枚重量相等,另一枚比它们或轻或重. 试用决策树描述一个算法,使得仅用一个天平且最多进行两次比较就可以确定出坏币(不必确定它是轻是重).

25. 根据图 3-27,使用哈夫曼码对每个二进制串进行解码.

(1)011000010　　　(2)01110100110

(3)01111001001110　　　(4)1110011101001111

26. 根据图 3-27,使用哈夫曼码对下列单词进行解码.

(1)den　(2)need　(3)leaden　(4)penned

27. 对表 3.1 中的字符集构造一个最优哈夫曼码.

表 3.1

字符	频率
α	5
β	6
γ	6
δ	11
ζ	20

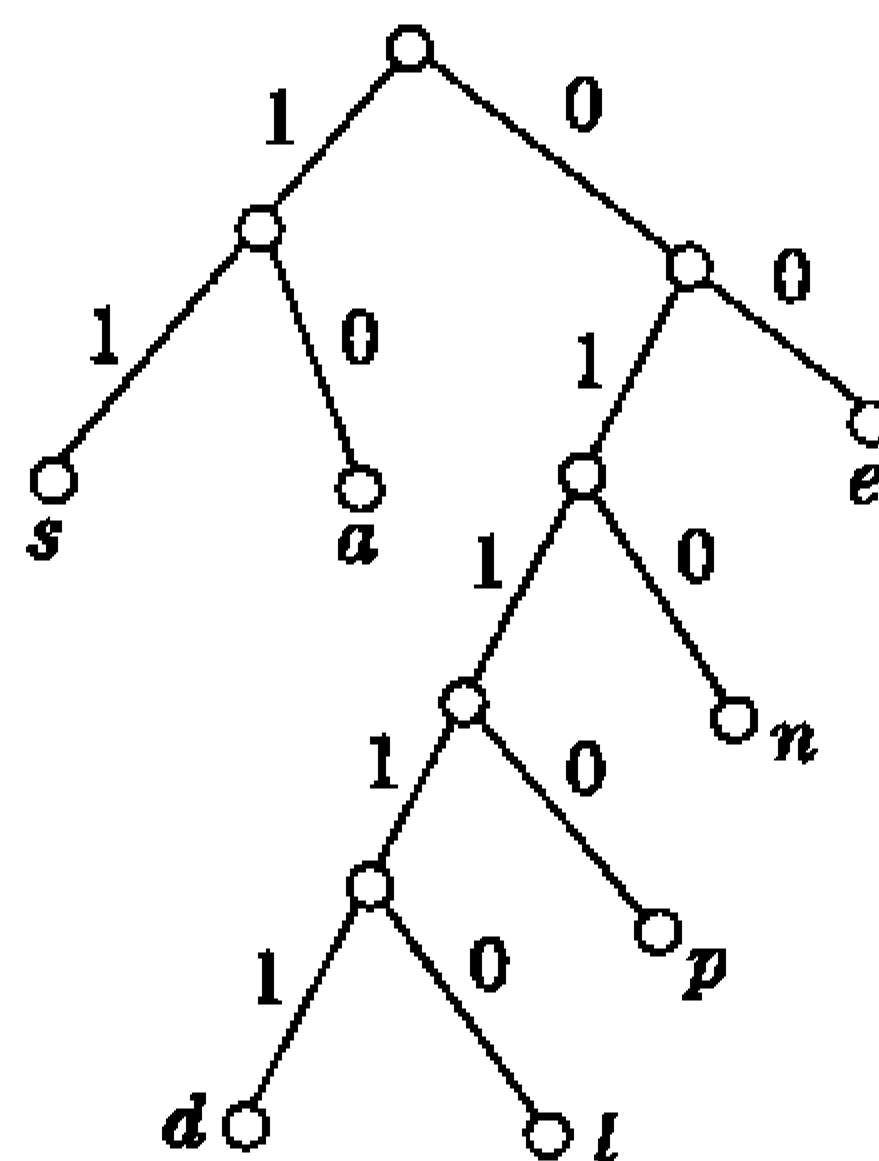


图 3-27

28. 编程解答下列问题:

(1)显示全部的互不同构的带有 7 个顶点的树.

(2)根据英文字母在普通英文资料中出现的频率,构造一种字母的哈夫曼码.

(3)求 n 个顶点的连通加权简单图的最小生成树.

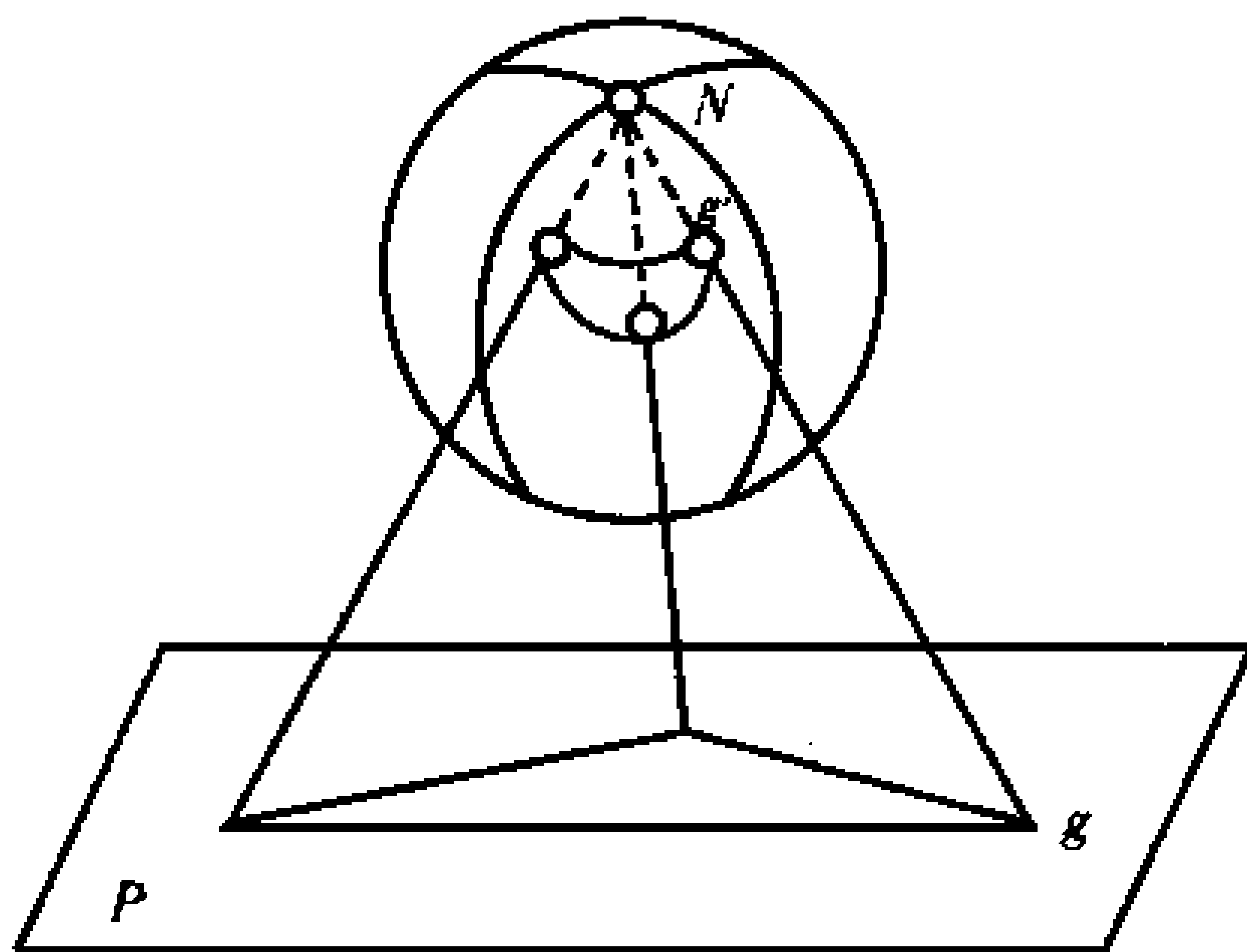
第四章　平面图

第一节　平面图

图的平面性问题,除了它的理论意义外,有许多实际应用.例如,单面印刷电路板和集成电路的布线问题.近年来,大规模集成电路的发展促进了图的平面性的研究.

定义 1　如果图 G 能够示画在曲面 S(如平面、球面、双环面等)上,且使得它的边仅在端点处相交,则称 G **可嵌入曲面 S(embeddable in the surface)**.如果图 G 可以嵌入平面上,则称 G 是**可平面图(plan graph)**,已经嵌入平面上的图 $\tilde{G}$ 称为 **G 的平面表示**.

可平面图 G 与 G 的平面表示 $\tilde{G}$ 是同构的.在不产生混淆的情况下,这两个概念通常不加以区别,都简称为平面图(**planar**



graph).

设 N 是球面的北极，平面 P 在球的下方，则平面上任一点 g 与 N 的连线必过球面上的唯一点 g'. 即球面上的点与平面上的点存在一一对应.

定理 1 图 G 可嵌入球面⇔图 G 可嵌入平面.

为此，我们只需讨论图的平面嵌入.

例 1 图 4-1(a)所示的 Q_3 是否可平面性?

解 Q_3 是可平面性的，它的一种平面表示如图 4-1(b).

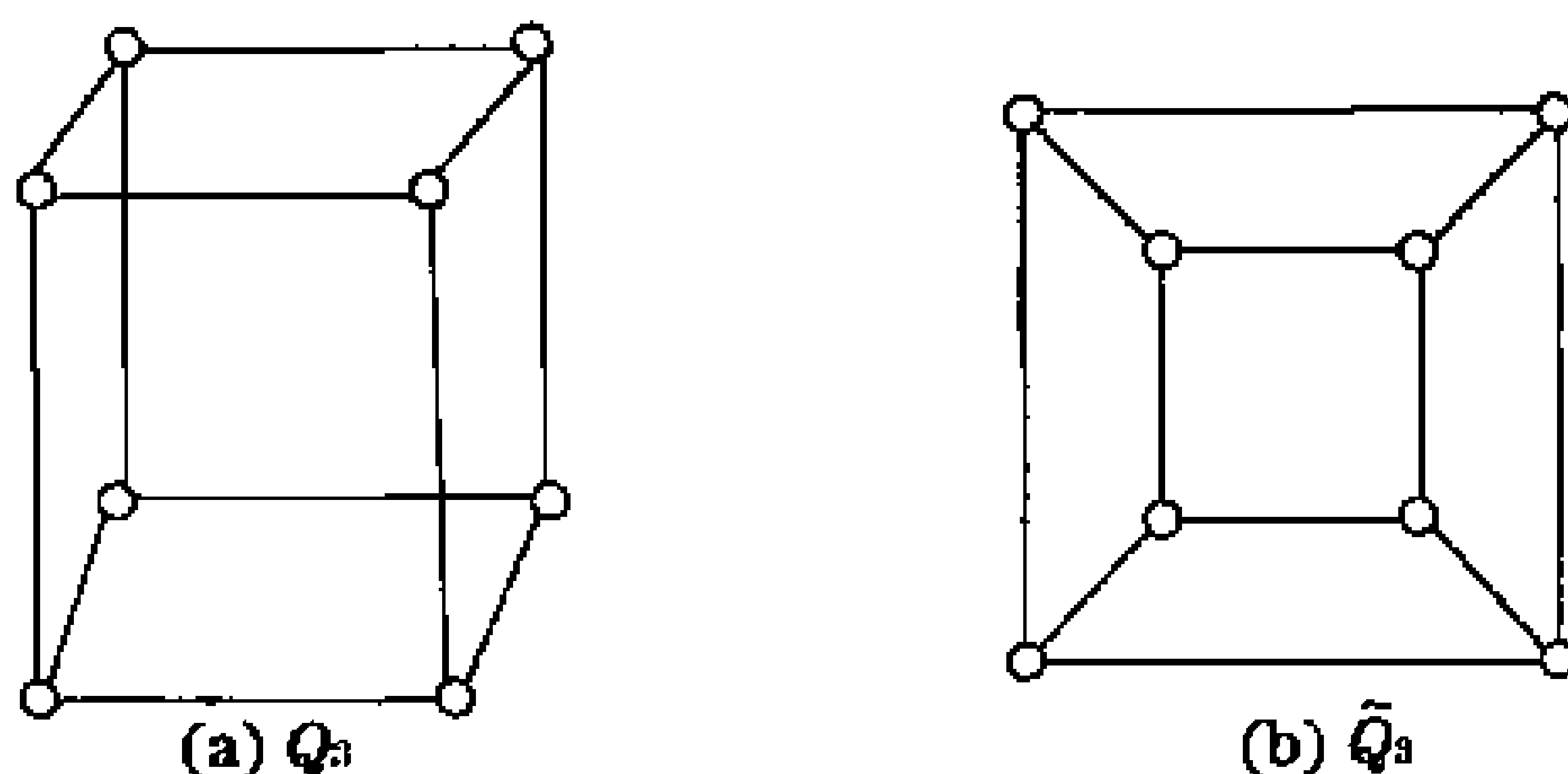


图 4-1

定义 2 设 G 是一个平面图，由 G 中的边所包围的区域，在区域内既不包含 G 的结点，也不包含 G 的边，这样的区域称为 G 的一个面(**face**). 有界区域称为内部面，无界区域称为**外部面**. 包围面的长度最短的闭链称为该面的**边界**(**boundary**). 面的边界的长度称为该**面的度数**(**degree of a face**).

例 2 指出图 4-2 所示平面图的面、面的边界，面的度数.

解 面 f_1，其边界 $1e_1 5e_2 4e_4 3e_7 2e_{10} 1$，$d(f_1)=5$

面 f_2，其边界 $1e_{10} 2e_8 7e_9 1$，$d(f_2)=3$

面 f_3，其边界 $2e_7 3e_6 7e_8 2$，$d(f_3)=3$

面 f_4，其边界 $3e_4 4e_5 7e_6 3$，$d(f_4)=3$

外部面 f_5，其边界 $1e_1 5e_2 4e_3 6e_3 4e_5 7e_9 1$，$d(f_5)=6$

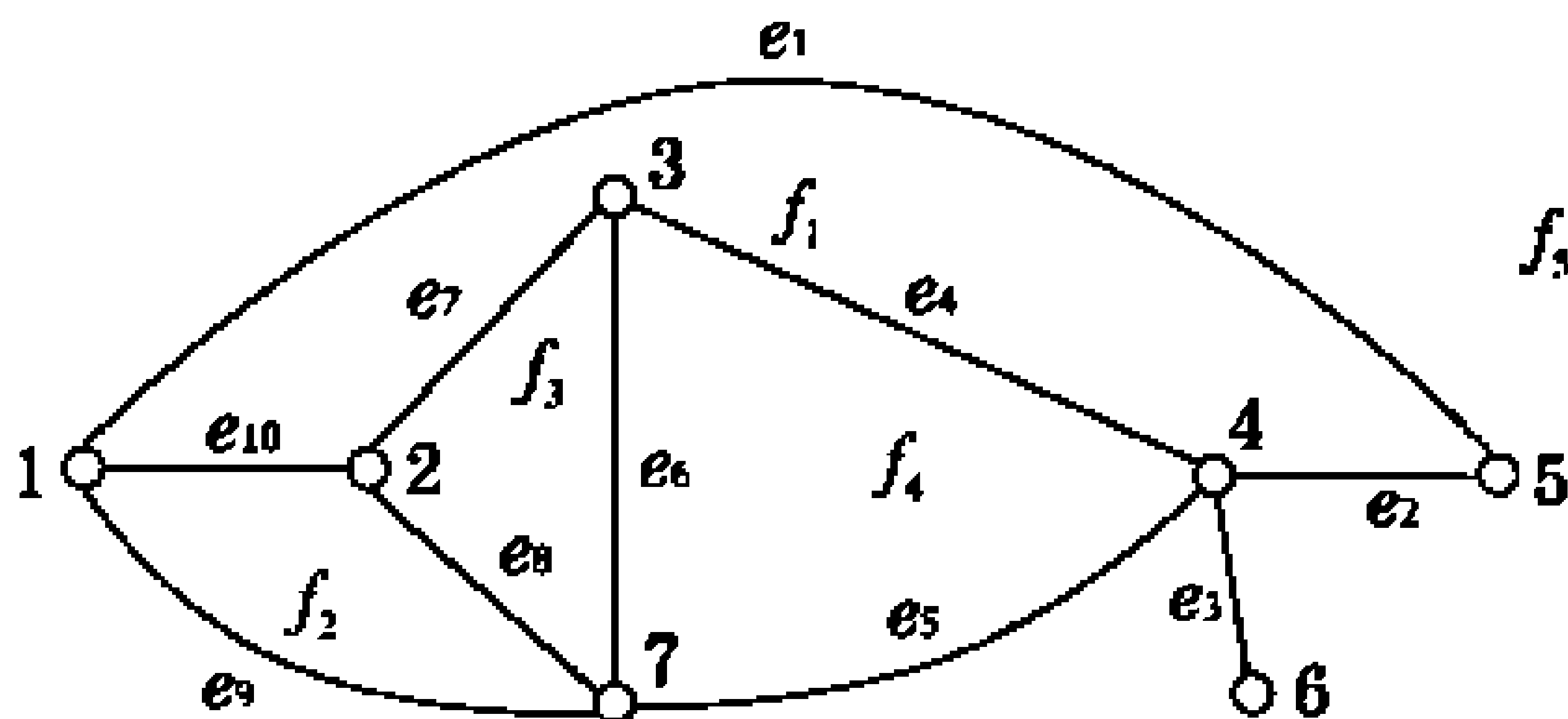
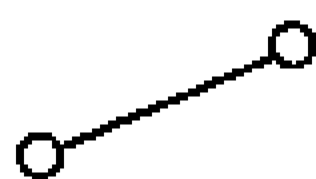


图 4-2

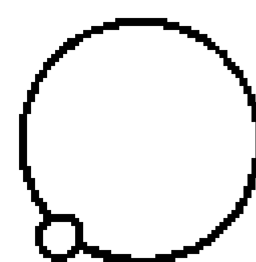
定理 2 对任何平面图 G,面的度数之和为边数的二倍.

定理 3(欧拉公式) 设 G 是带 e 条边,v 个顶点和 r 个面的平面图,则 $v-e+r=2$.

证明 (1)$e=1$ 时,图 G 有两种情形,如图 4-3 所示,其中任一种情况都满足公式.



$v=2 \quad e=1 \quad r=1$



$v=1 \quad e=1 \quad r=2$

图 4-3

(2)假设公式对有 n 条边的图成立.设 G 有 $n+1$ 条边.

若 G 不含圈,任取一点 x,从结点 x 开始沿路行走.因 G 不含圈,所以每次沿一边总能达到一个新结点,最后我们会到达一个度数为 1 的结点,不妨设为 a,在结点 a 不能再继续前进.删除结点 a 及其关联的边得图 G',G' 含有 n 条边.由假设公式对 G' 成立,而 G 比 G' 多一个结点和一条边,且 G 与 G' 面数相同,故公式也适用于 G.

若 G 含有圈 C,设 y 是圈 c 上的一边,则边 y 一定是两个不同面的边界的一部分.这时,我们删除边 y 得图 G',则 G' 有 n 条边,由假设公式对 G' 成立.而 G 比 G' 多一边和多一面,G 与 G' 的

顶点数相同,故公式也适合于 G.

(3)由(1)(2)及数学归纳法知定理成立.

推论 1 设 G 是带 e 条边和 v 个顶点的连通平面简单图,其中 $v\geqslant 3$,则 $e\leqslant 3v-6$.

证明 由于 G 是简单图,因此 G 的每个面的度数至少为 3. 所以,$2e=\sum d(f)\geqslant 3r$,其中 r 为 G 的面数. 由欧拉公式得 $e\leqslant 3v-6$.

推论 2 设 G 是带 e 条边和 v 个顶点的连通平面简单图,其中 $v\geqslant 3$ 且没有长度为 3 的圈,则 $e\leqslant 2v-4$.

证明 G 的每个面的度数至少为 4.

例 3 K_5 和 $K_{3,3}$均是非平面图.

解 图 K_5 有 5 个顶点 10 条边,而 $3\times 5-6=9$,即 $10>9$,故由推论 1 知,K_5 是非平面图.

图 $K_{3,3}$没有长度为 3 的圈,且有 6 个顶点 9 条边,因而 $9>2\times 6-4$,故由推论 2 知,$K_{3,3}$是非平面图.

推论 3 设 G 是带 e 条边、v 个顶点和 r 个面的平面图,则 $v-e+r=1+w$,其中 w 为连通分支数.

推论 4 设 G 是任意平面图,则 $\delta(G)\leqslant 5$.

证明 设 G 有 v 个顶点 e 条边. 若 $e\leqslant 6$,结论显然成立;若 $e>6$,假设 G 的每个顶点的度数$\geqslant 6$,则 $6v\leqslant \sum\limits_{v_i\in V(G)} d(v_i)=2e\leqslant 6v-12$,因而假设不成立,即 $\delta(G)\leqslant 5$.

例 4 平面上有 n 个点,其中任两个点之间的距离至少为 1. 证明在这 n 个点中,距离恰好为 1 的点对数至多是 $3n-6$.

证明 首先建立图 $G=\langle V,E\rangle$,其中 V 就取平面上给定的 n 个点(位置也不变),两个点之间的距离为 1 时,该两顶点之间用一条直线段连接. 显然图 G 是一个 n 阶简单图. 我们只要证明 G 是

平面图即可.

若 G 中存在两条不同的边 $\{a,b\}$ 和 $\{x,y\}$ 相交于非端点处 o，如图 4-4(a)所示，记它们的夹角为 $\theta(0<\theta<\pi)$.

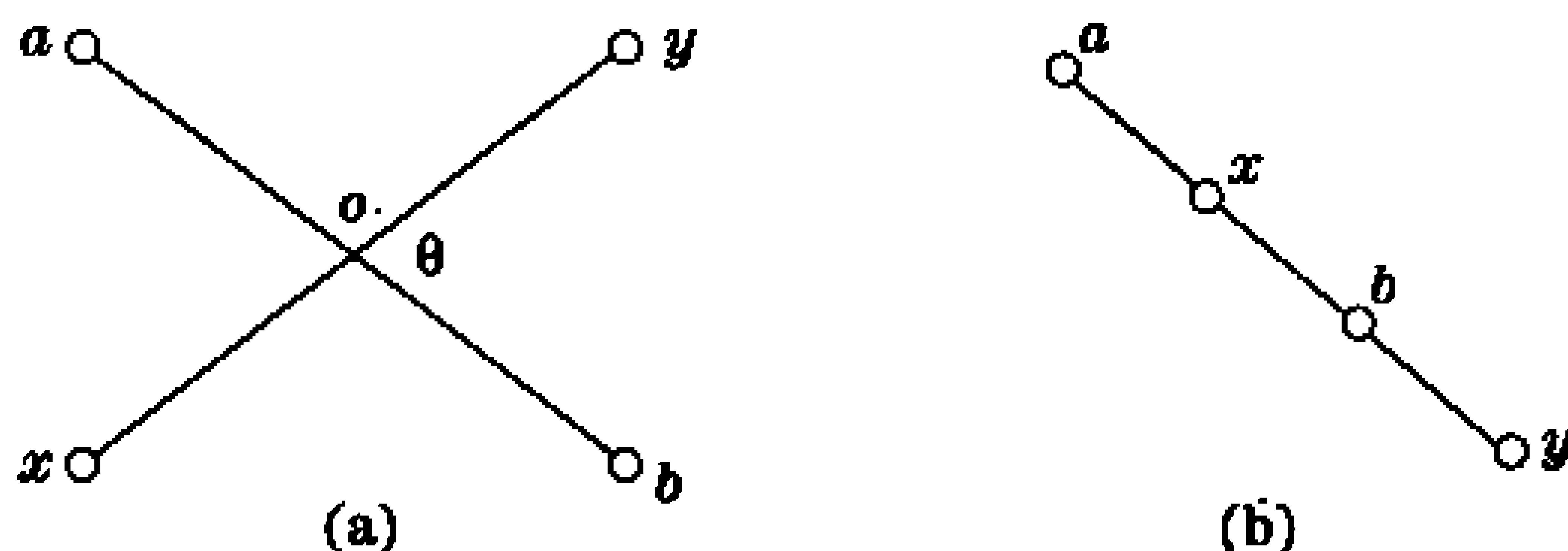


图 4-4

若 $\theta=\pi$，这时情形如图 4-4(b)所示，存在两点其距离小于 1，这是不可能的，因而 $0<\theta<\pi$. 由于 a 到 b 的距离为 1，x 到 y 的距离为 1，a,b,x,y 中至少有两个点，从交点 o 到这两点的距离不超过 $\frac{1}{2}$，不妨设为 a,x，则点 a 与 x 之间的距离小于 1，矛盾.

因此 G 是平面图，即命题得证.

第二节　库拉图斯基定理与极大平面图

定义 1　设 G 是一个平面图，通过删除 G 的一条边 $\{x,y\}$，并且添加一个新结点 a 和两条边 $\{x,a\}$ 与 $\{a,y\}$（所获得的任何图也是平面图），这样的操作称为**初等细分**. 若可以从相同的图 G 通过一系列初等细分来获得图 G_1 和 G_2，称 G_1 和 G_2 是**同胚的**(**homeomorphism**).

如图 4-5 所示的三个图 G_1,G_2,G_3 都是同胚的.

波兰数学家库拉图斯基(Kazimierz kuratowski, 1896～1980)在 1930 年证明了下面的定理，这个定理用图同胚的概念刻画了平面性图.

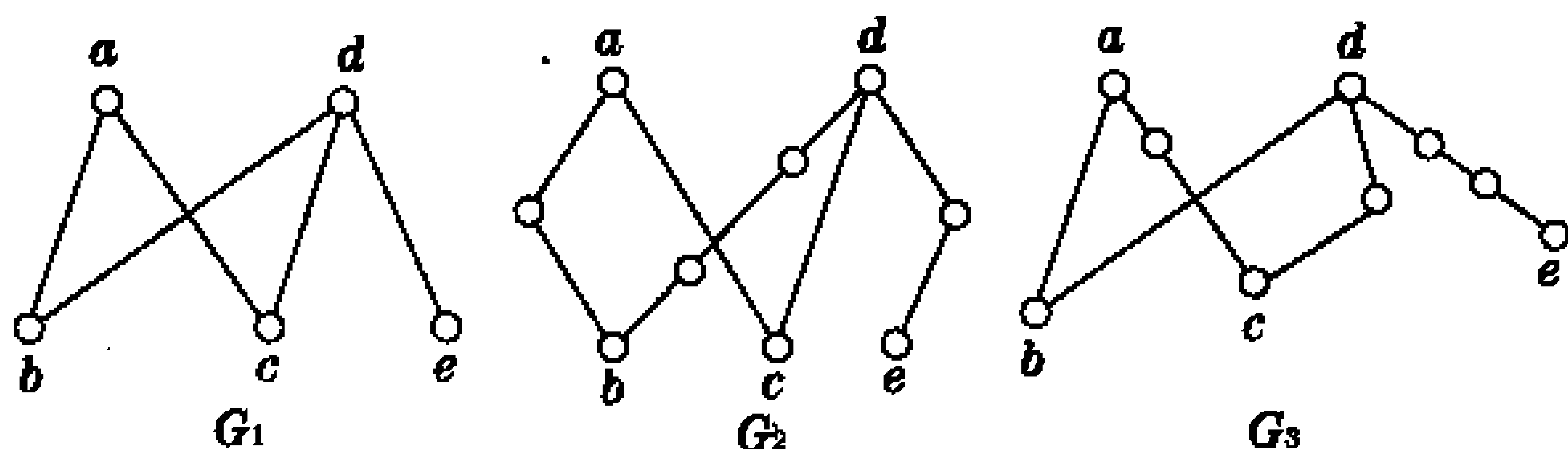


图 4-5

定理 1 一个图是非平面的,当且仅当它包含一个同胚于 $K_{3,3}$ 或 K_5 的子图.

证明略.

例 1 说明彼得森图不是平面图.

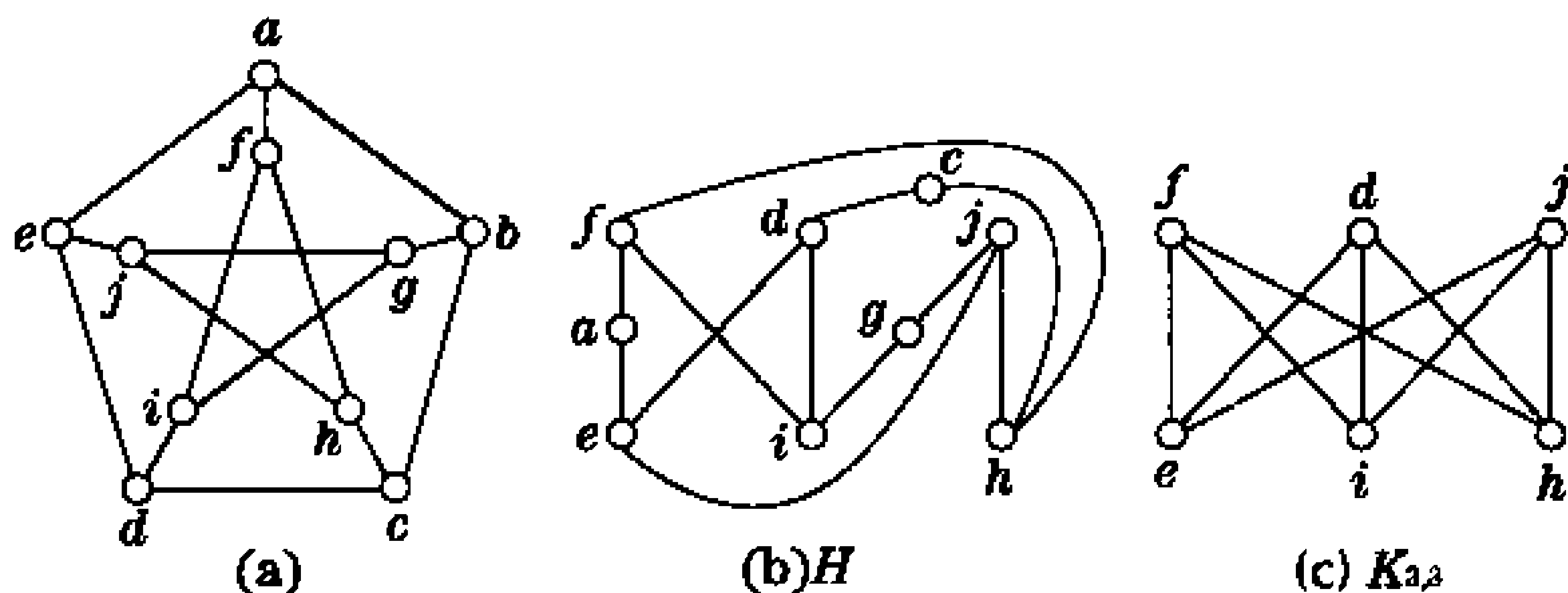


图 4-6

解 删去图 4-6(a)彼得森图的结点 b,得到它的一个子图为图 4-6(b)所示 H.而 H 同胚于 $K_{3,3}$,故彼得森图不是平面图.

显然,库拉图斯基给出了平面图的充要条件,但用它并不能得出判别一个图是否平面图的有效算法.

下面介绍极大平面图.

定义 2 设 G 是阶大于等于 3 的简单可平面图,若在任意两个不相邻的结点 v_i, v_j 之间加入边 $\{v_i, v_j\}$,就会破坏图的平面性,则称 G 是**极大平面图**(**maximal planar graph**).极大平面图的

平面表示称为**三角剖分平面图**(**triangulation of a planar graph**)，简称为**三角剖分图**(**triangulation**)．若 G 是极大平面图，显然 G 是连通的，且 G 中不存在割边．$v(\geqslant 3)$阶简单平面图 G 是极大平面图的充要条件：

(1)G 中每个面的度数都是 3.

(2)设 G 有 ε 条边 r 个面，则 $3r=2\varepsilon$.

(3)设 G 带有 v 个顶点、ε 条边和 r 个面，则 $\varepsilon=3v-6$，$r=2v-4$.

证明(1)，因为 G 为简单图，所以 G 中没有环和平行边，因此不存在度数为 1 或 2 的面．假设 G 存在度数大于 3 的面 f，不妨设其为内部面，如图 4-7 所示，显然 v_1 与 v_3 不相邻，在面 f 内加入边 $\{v_1,v_3\}$，不改变图 G 的平面性，这与图 G 是极大平面图矛盾．充分性显然．

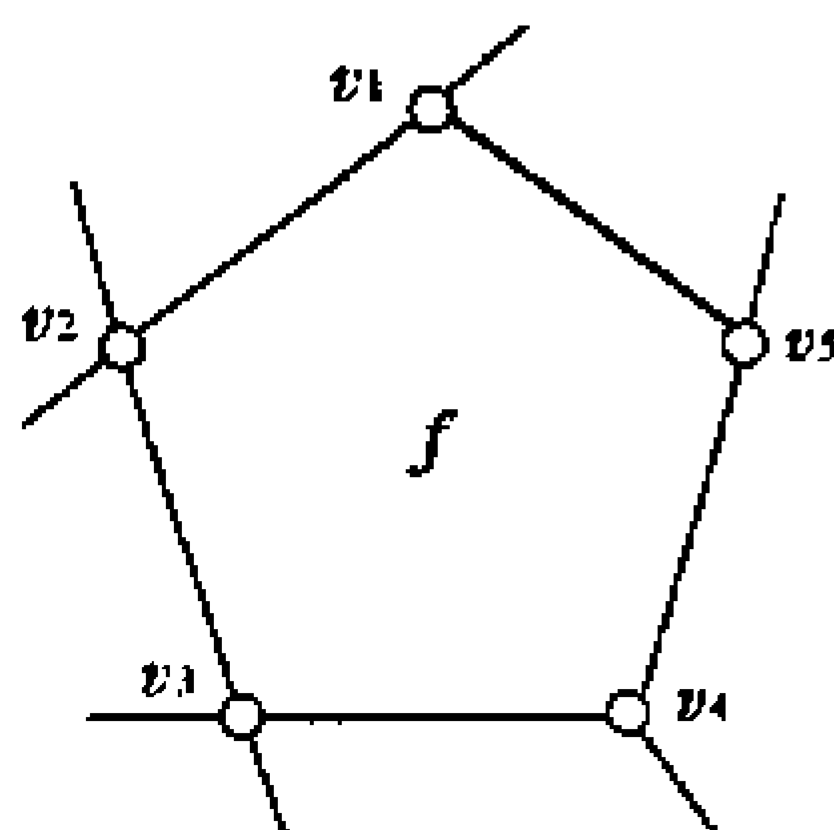


图 4-7

定理 2　设 G 是简单平面图，则 G 有平面表示 $\tilde{G}$，使 $\tilde{G}$ 中每条边都是直线．

证明　只要对极大平面图 G 来证明定理成立即可．对阶数 $v(\geqslant 3)$用归纳法．当 $v=3$ 时，G 是三角形，定理显然成立．假设定理对所有阶数小于 v 的极大平面图成立，并设 G 是 $v(\geqslant 4)$阶极大平面图．不妨设 G 是三角剖分图．选取 $x\in V(G)$使 x 不是外部面边界上的点．取边 $\{x,y\}$．于是边 $\{x,y\}$ 仅是某两个内部三角

形的公共边.不妨设这两个三角形分别为 z_1xy 和 z_2xy(见图 4-8(b)).收缩边 $\{x,y\}$,且结点 x 和 y 收缩为 P,得图 G'(见图 4-8(c)).显然 G' 是平面图,而且有 $|E(G')|=|E(G)|-3=3(|V(G)|-1)-6=3|(V(G)|-9=3|V(G')|-6$,因此 G' 是 $v-1$ 阶极大平面图.由归纳假设,G' 有平面表示 $\tilde{G}'$,使 $\tilde{G}'$ 的每条边都是直线.

考虑 $\tilde{G}'$ 中边 Pz_1 和 Pz_2,将它们分裂成两个三角形(见图 4-8(b)和(c)).这样得到的图 $\tilde{G}$ 就是 G 的平面表示,而且每条边都是直线段.定理得证.

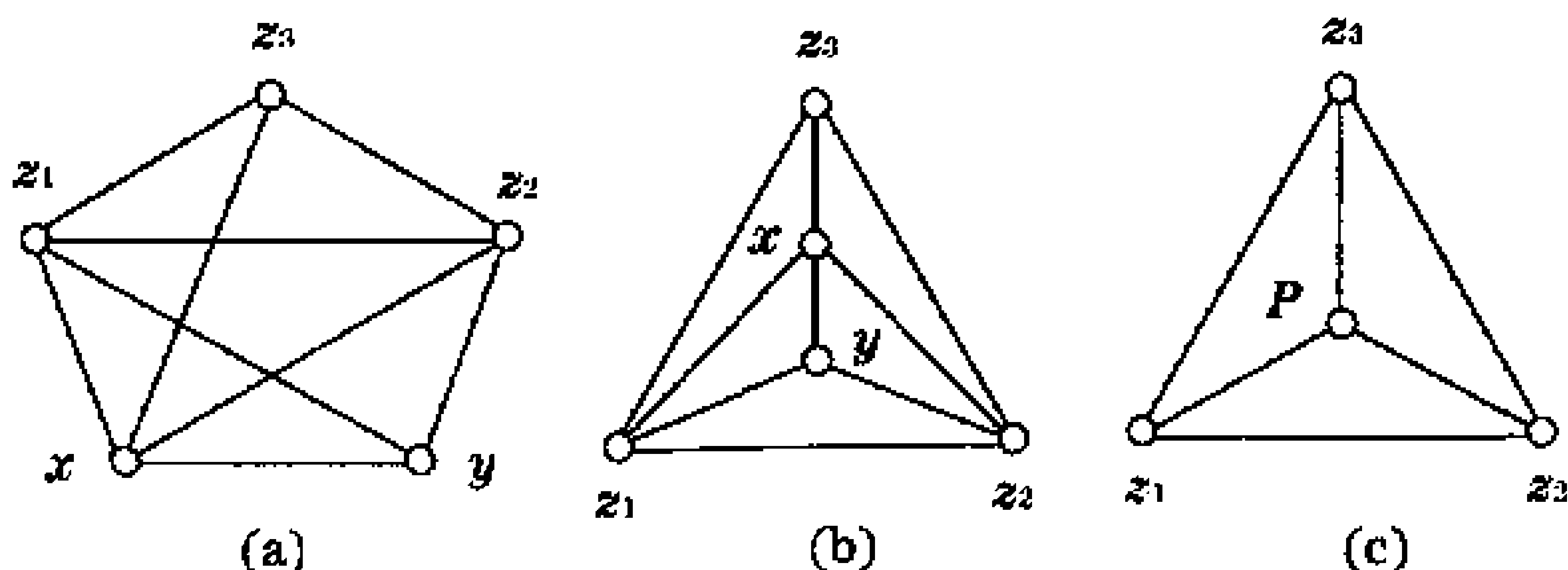


图 4-8

例如,图 4-9(b)是图 4-9(a)所示图的平面直线段表示.

最后,我们来介绍正多面体.

平面图的理论与多面体的研究密切相关:事实上,由于每个凸多面体 P 可以与一个连通可平面图 G 相对应,G 的顶点和边是 P 的顶点和棱,那么 G 的每个顶点的度至少为 3.由于 G 是一个平面图,那么 P 的面就是 G 的面,并且 G 的每一条边落在两个不同面的边界上.图 4-10 显示了凸多面体和它相应的平面图.

一个多面体 P 的顶点、棱和面的数目分别用 V,E 和 F 来表示,而且,这些分别是连通平面图 G 的顶点、边和面的数目.因此,欧拉公式可写成 $V-E+F=2$,这就是著名的 **Euler 凸多面体公**

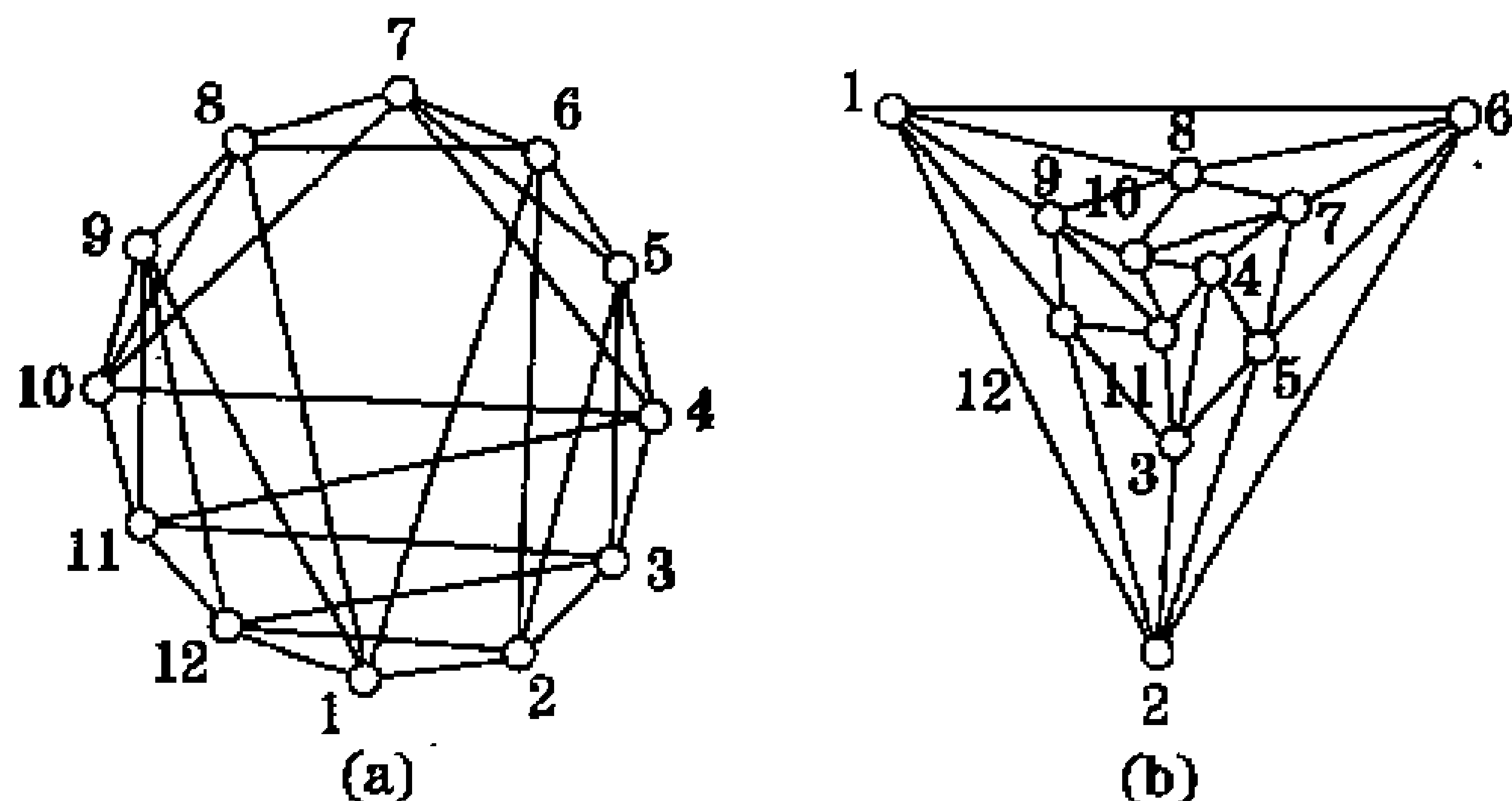


图 4-9

式.

为方便起见,用 V_n 和 F_n 分别表示凸多面体 P(或对应的平面图 G)的 n 度点和 n 度面的数目.于是 $n\geqslant3$,且有

$$2E=\sum_{n\geqslant3} nV_n=\sum_{n\geqslant3} nF_n$$

定理 3 每个凸多面体中都至少有一个 n 度面,其中 $3\leqslant n\leqslant5$.

证明 假设 $F_3=F_4=F_5=0$,则

$$2E=\sum_{n\geqslant6} nF_n\geqslant\sum_{n\geqslant6} 6F_n=6\sum_{n\geqslant6} F_n=6F$$

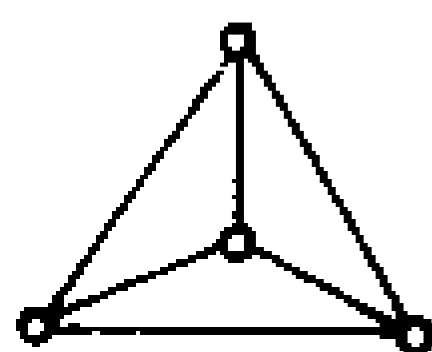
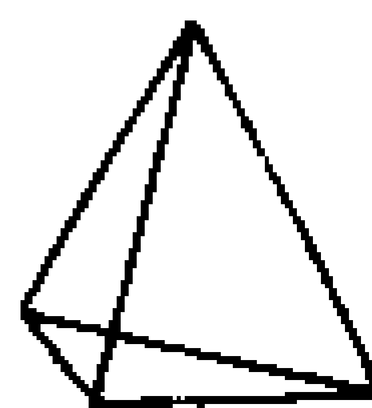
即有
$$F\leqslant\frac{1}{3}E$$

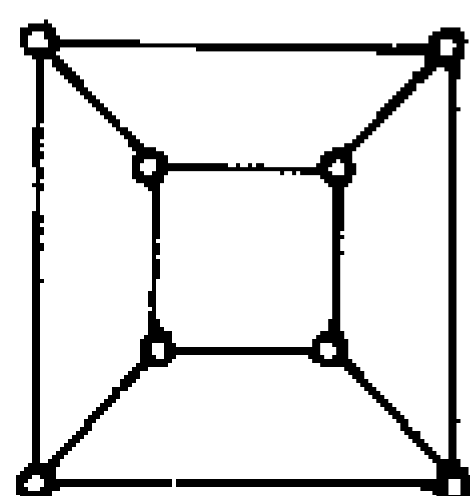
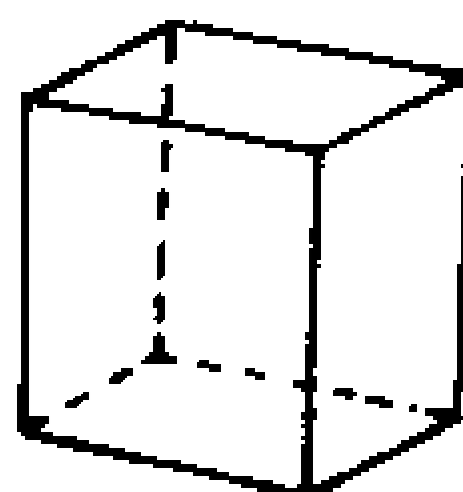
另一方面,有 $2E=\sum_{n\geqslant3} nV_n\geqslant3\sum_{n\geqslant3} V_n=3V$

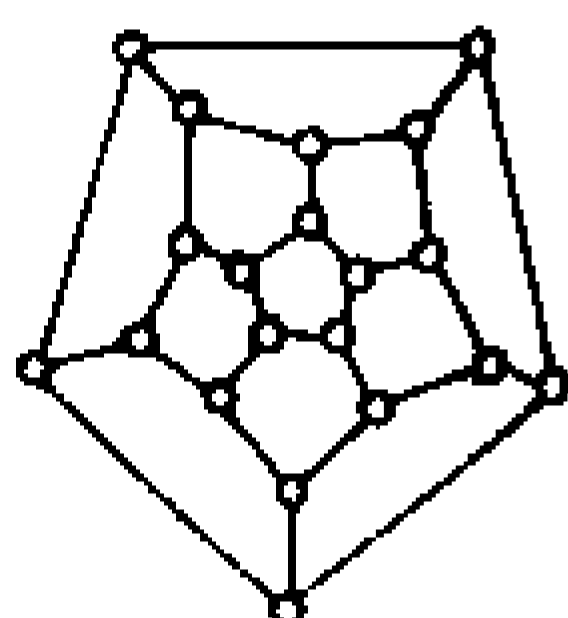
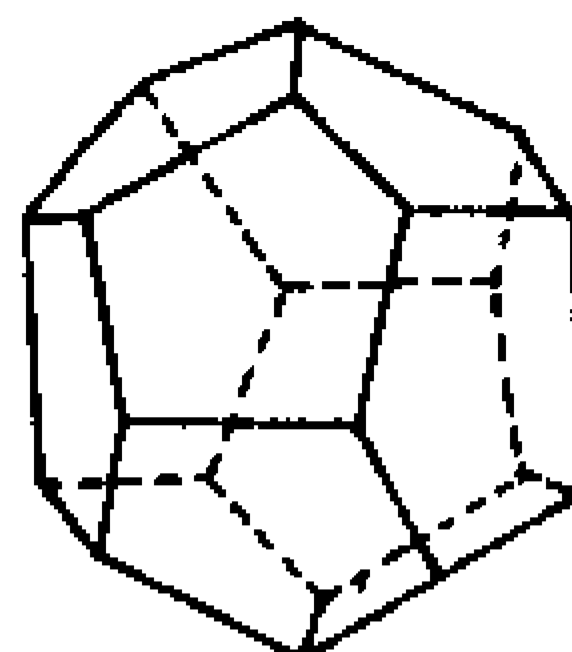
即
$$V\leqslant\frac{2}{3}E$$

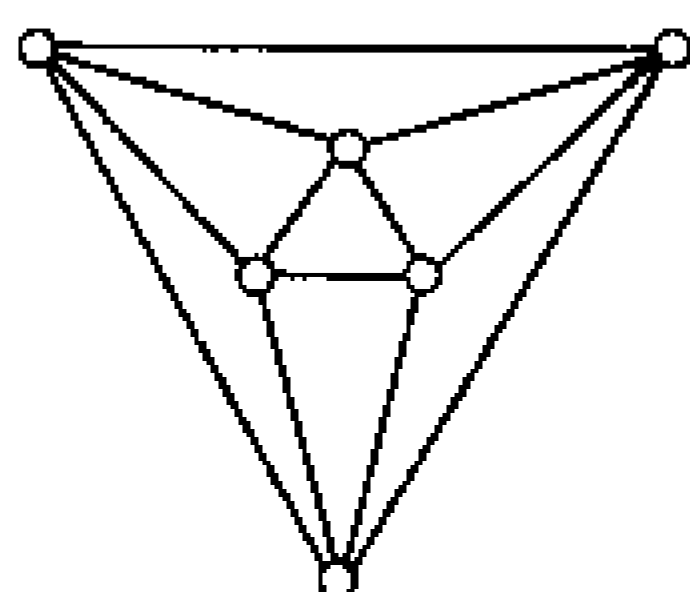
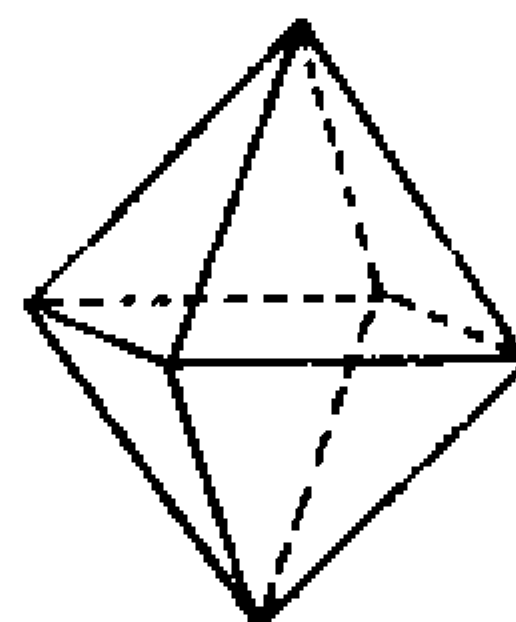
于是得到下列矛盾,$E=V+F-2\leqslant\frac{2}{3}E+\frac{1}{3}E-2=E-2$

每个面并且每个多面角都相等的凸多面体称为**正多面体**(**regular polyhedron**),也称为 **Plato 体**.对应的平面图可称为 Plato


四面体



立方体



十二面体



八面体


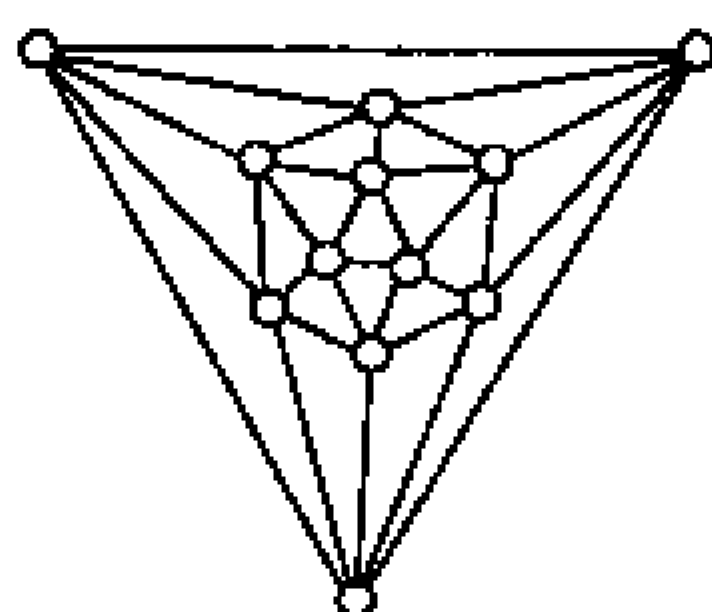
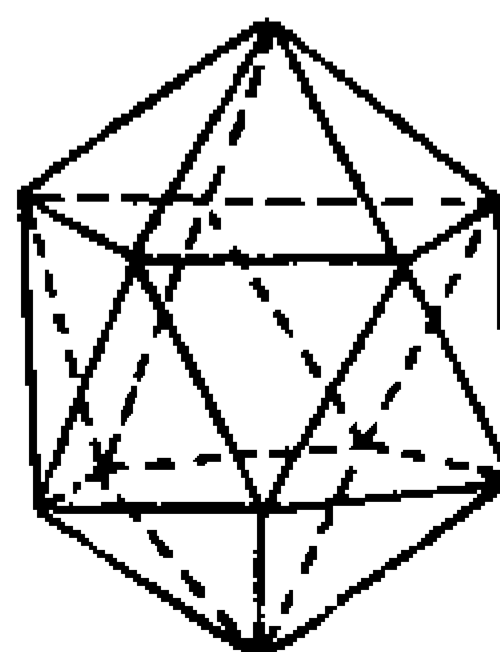

二十面体


图 4-10

图.

定理 4 仅有五个正多面体(如图 4-10 所示).

证明　设 P 是正多面体且 G 是对应的平面图，对任何凸多面体，均有

$$\begin{aligned}-8&=4E-4V-4F\\&=2E+2E-4V-4F\\&=\sum_{n\geqslant 3}nF_n+\sum_{n\geqslant 3}nV_n-4\sum_{n\geqslant 3}V_n-4\sum_{n\geqslant 3}F_n\\&=\sum_{n\geqslant 3}(n-4)F_n+\sum_{n\geqslant 3}(n-4)V_n\end{aligned}$$

因为 P 是正多面体，所以存在两个整数 $h(\geqslant 3)$ 和 $k(\geqslant 3)$ 使 $F=F_h$ 且 $V=V_k$，因此，有 $-8=(h-4)F_h+(k-4)V_k$，且 $hF_h=2E=kV_k$，由于 $3\leqslant h\leqslant 5$.

(1)当 $h=3$ 时，有 $12=(6-k)V_k$，由于 $3\leqslant k\leqslant 5$.

当 $k=3$ 时，$V_3=4$，$F_3=4$，因此 P 是四面体；

当 $k=4$ 时，$V_4=6$，$F_3=8$，因此 P 是八面体；

当 $k=5$ 时，$V_5=12$，$F_3=20$，因此 P 是二十面体.

(2)当 $h=4$ 时，有 $8=(4-k)V_k$，所以 $k=3$，$V_3=8$，$F_4=6$，即 P 是立方体.

(3)当 $h=5$ 时，有 $20=(10-3k)V_k$，所以 $k=3$，$V_3=20$，$F_5=12$，即 P 是十二面体.

例 2　对哪些 n，存在 n 条棱的凸多面体？

解　以多面体的顶点为图的顶点，以多面体的棱为图的边，得到一个平面图 G，若用 $p(G)$，$q(G)$，$\varphi(G)$分别表示 G 的顶点数，边数和面数，则 $p(G)\geqslant 4$，$\varphi(G)\geqslant 4$，且每个面的度数至少是3. 由 Euler 公式，$q(G)\geqslant 6$，即没有棱数小于 6 的凸多面体.

四面体是棱数为 6 的凸多面体. 若有 7 条棱的凸多面体，则存在满足上述条件，$q(G)=7$的平面图，由 Euler 公式得

$$p(G)+\varphi(G)=q(G)+2=9\tag{1}$$

但 G 的每个面的度至少是 3，故

$$2q(G)=\sum d_G(f)\geqslant 3\varphi(G)(f\text{ 为 }G\text{ 的面})$$

即
$$\varphi(G)\leqslant\frac{2}{3}q(G)=\frac{14}{3}$$

但 $\varphi(G)$为整数,所以 $\varphi(G)\leqslant 4$.同样 $p(G)\leqslant 4$,于是 $p(G)$ + $\varphi(G)\leqslant 8$,这与式(1)矛盾.所以 7 条棱的凸多面体不存在.

现对 $k\geqslant 4$,以 k 边形为底的棱锥即为 $2k$ 条棱的凸多面体.若把底为 $k-1$ 边形的棱锥中,底角处的一个三面角"锯掉一个尖儿",得到的是一个有 $2k+1$ 条棱的凸多面体,总之,当 $n\geqslant 6$ 且 $n\neq 7$时,才有 n 条棱的凸多面体.

第三节　图的平面性检测

设 H 是G 的子图,在 $E(G)-E(H)$上定义二元关系 R 如下:$xRy\Leftrightarrow$在 $G-E(H)$中存在一条链 W 使得

(1)W 的第一条边为 x,最后一条边为 y;

(2)W 只有两个端点属于H(即 W 的内部点不属于H).

例如,图 4-11 所示图 G 的子图H 即图 4-11 中的虚线所示.

记$A=E(G)-E(H)$
$=\{e_1,e_2,e_3,e_4\}$

$$\begin{aligned}A\times A=\{&\langle e_1,e_1\rangle\langle e_1,e_2\rangle\langle e_1,e_3\rangle\langle e_1,e_4\rangle\\&\langle e_2,e_1\rangle\langle e_2,e_2\rangle\langle e_2,e_3\rangle\langle e_2,e_4\rangle\\&\langle e_3,e_1\rangle\langle e_3,e_2\rangle\langle e_3,e_3\rangle\langle e_3,e_4\rangle\\&\langle e_4,e_1\rangle\langle e_4,e_2\rangle\langle e_4,e_3\rangle\langle e_4,e_4\rangle\}\end{aligned}$$

则 $R=\{\langle e_1,e_1\rangle\langle e_2,e_2\rangle\langle e_2,e_3\rangle\langle e_2,e_4\rangle\langle e_3,e_2\rangle\langle e_3,e_3\rangle\langle e_3,e_4\rangle$
$\langle e_4,e_2\rangle\langle e_4,e_3\rangle\langle e_4,e_4\rangle\}$

很容易证明,上述定义的二元关系 R 是等价关系.由等价关系与划分的关系知,R 确定$E(G)-E(H)$上的一个划分 S,不妨

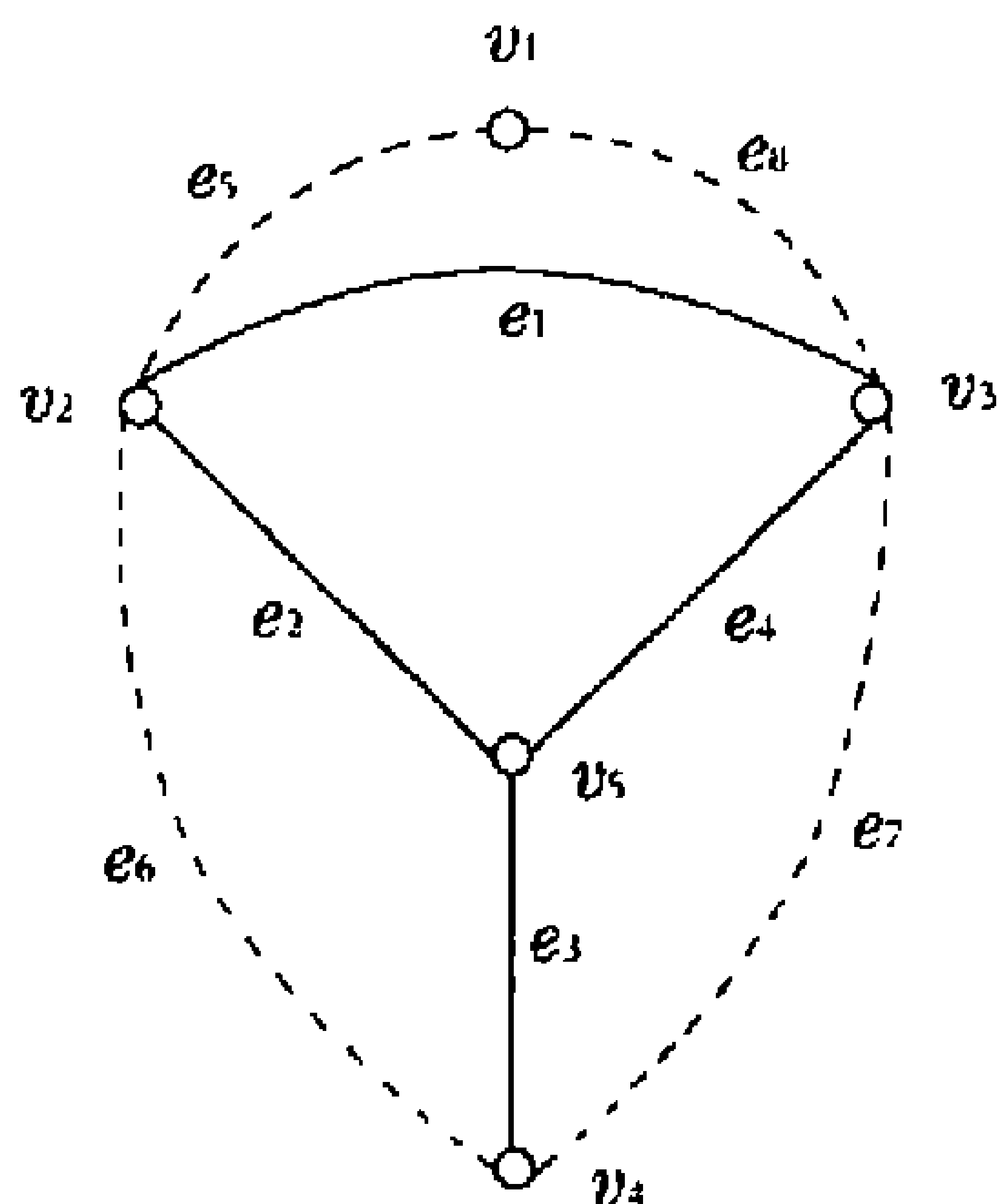


图 4-11

设 $S=\{S_1,S_2,\cdots,S_m\}$,则由 S_i 导出的 $G-E(H)$ 中的子图 B_1, $B_2,\cdots,B_m$ 称为 G 的 ***H* 片(piece)**.

上例中,由 R 确定的 $A=E(G)-E(H)=\{e_1,e_2,e_3,e_4\}$ 的划分 $S=\{[e_1],[e_2],[e_3],[e_4]\}=\{\{e_1\},\{e_2,e_3,e_4\}\}$. 因而图 4-11 所示的 $G-E(H)$ 的片为图 4-12 所示的 B_1 和 B_2.

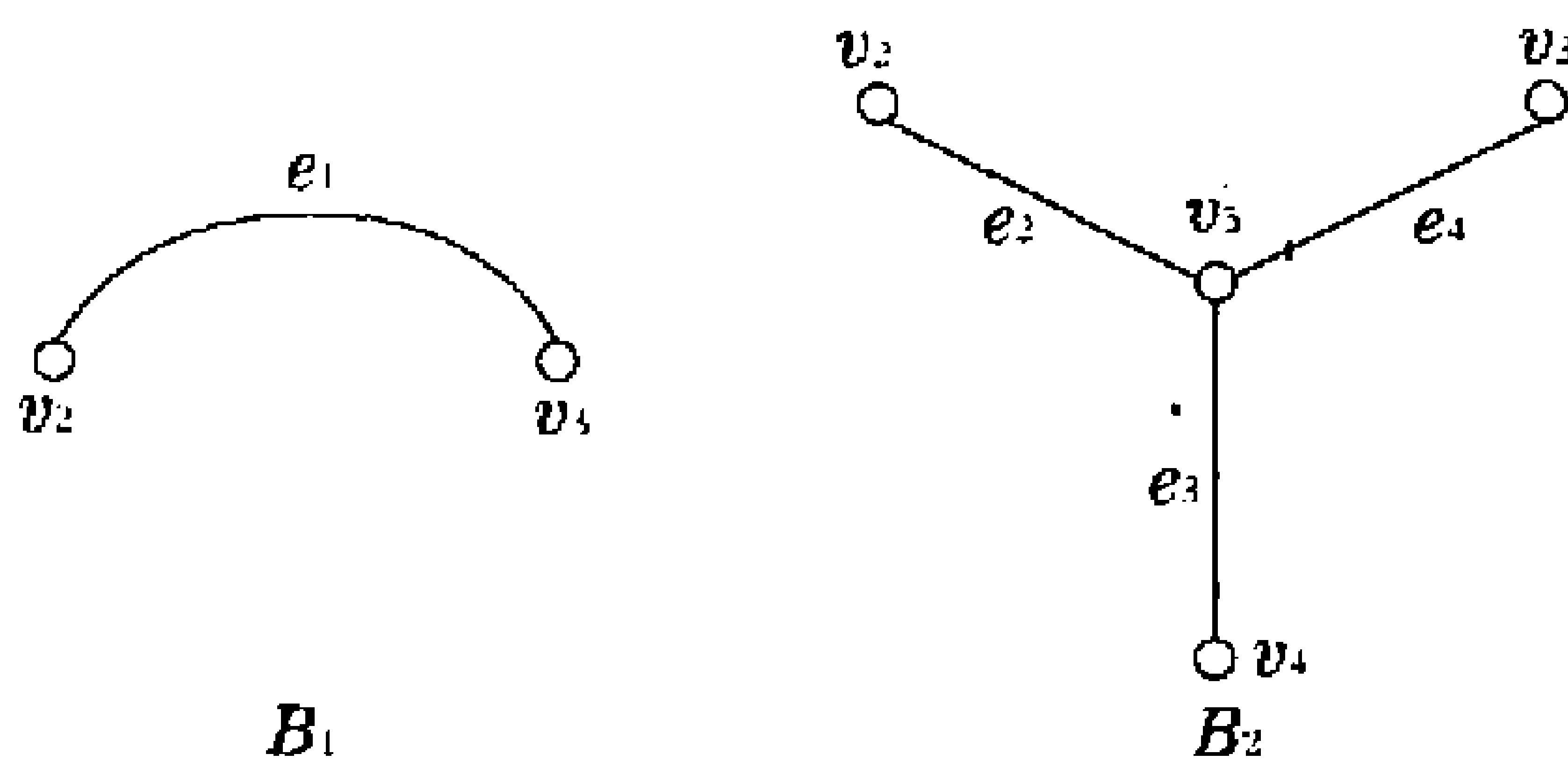


图 4-12

若 H_1 和 H_2 都是图 G 的子图,称 $V(H_1)\cap V(H_2)$ 为 H_1 和 H_2 在 G 中的**接触点集(vertices of attachment)**,,记作 $V_G(H_1$,

H_2).

设 H 是可平面图 G 的子图，$\widetilde{H}$ 是 H 的平面表示. 若存 G 的平面表示 $\widetilde{G}$ 使 $\widetilde{H}\subseteq\widetilde{G}$，称 $\widetilde{H}$ 是 **G 容许的**.

例如，图 4-13 表示 G 的一个可平面子图的两种嵌入：一个是 G 容许的，另一个则不是.

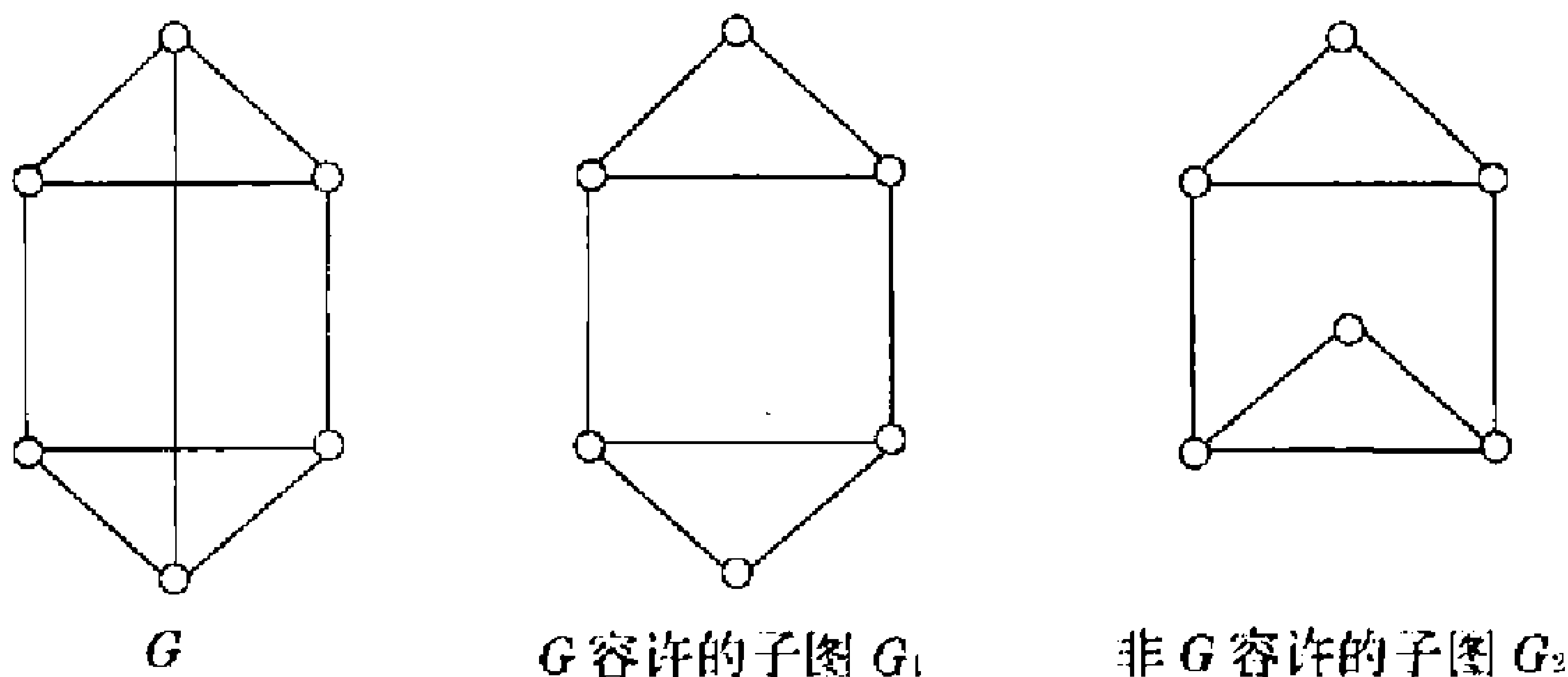


图 4-13

若 B 是 G 的 H 片，f 是 $\widetilde{H}$ 的面并且 $V_G(B,H)\subseteq B_{\widetilde{H}}(f)$，称 B 在 f **内可画出**. 其中 $B_{\widetilde{H}}(f)$ 是面 f 的界.

例如，图 4-14 所示为图 4-11 中的 H，H 的片如图 4-12，则

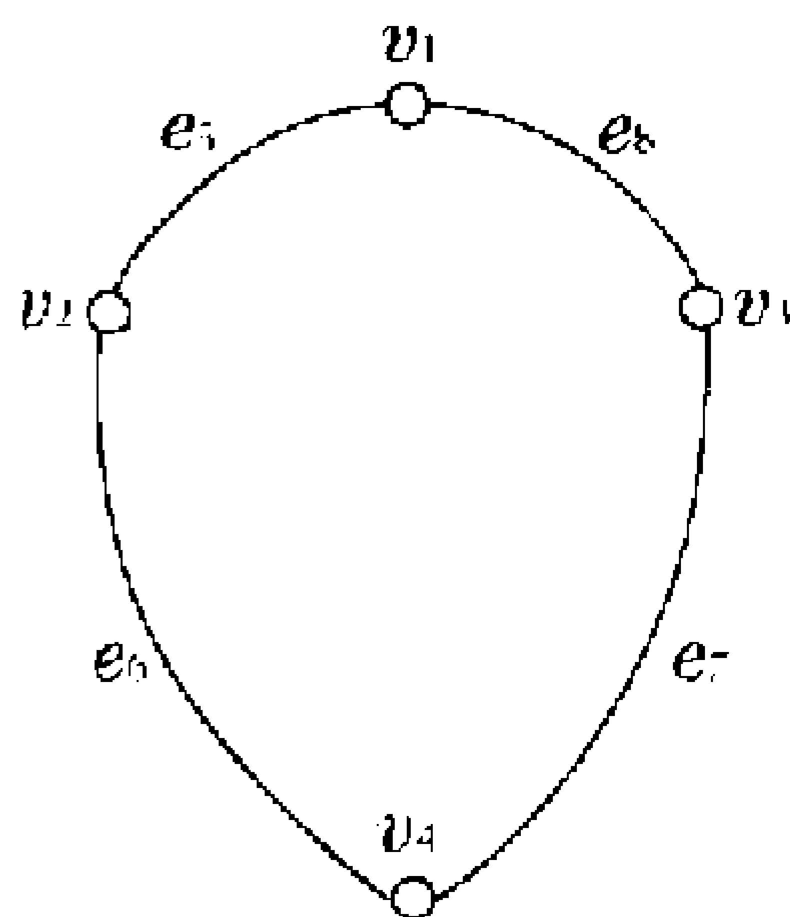


图 4-14

$$B_H(f)=v_1e_5v_2e_6v_4e_7v_3e_8v_1$$

$$V_G(B_1,H)=\{v_2,v_3\}$$
$$V_G(B_2,H)=\{v_2,v_3,v_4\}$$

故 B_1 和 B_2 都在 f 内可画出.

定理 1 设 $\widetilde{H}$ 是 G 容许的,则对 $\widetilde{H}$ 的每一个片 B,$F_G(B,\widetilde{H})\neq\varnothing$,其中 $F_G(B,\widetilde{H})=\{f\mid F(\widetilde{H})$ 为 $\widetilde{H}$ 的面集,$f\in F(\widetilde{H})$,且 B 在 f 内可画出$\}$

证明 若 $\widetilde{H}$ 是 G 容许的,则存在 G 的一个平面表示 $\widetilde{G}$,使 $\widetilde{H}\subseteq\widetilde{G}$.显然,$H$ 的片 B 所对应的 $\widetilde{G}$ 的子图必然限制在 $\widetilde{H}$ 的一个面中,因此 $F(B,\widetilde{H})\neq\varnothing$.证毕

下面我们来介绍平面检测的 *DMP* 算法,这是 Demoucron, Malgrange, Pertuiset 提供的.在介绍这个算法之前,为简化运算量,我们先对图 G 进行如下预处理:

(1)若 G 不连通,分别检测每一个连通分支.当且仅当所有的连通分支都是平面图,G 就是平面图.

(2)若 G 有割点,分别检测每一块.当且仅当每一块都是平面图,G 就是平面图.

(3)删去 G 中的环.

(4)用一条边代替 G 中 2 度点和与之相关联的两条边.

(5)删去平行边.

反复交错使用(4)与(5),直到不能使用为止.在做了上述简化后,在简单图 G 中利用(6)与(7)两个基本判断法.

(6)若 $e<9$ 或 $v<5$(e 为边数,v 为顶点数),则 G 是平面图.

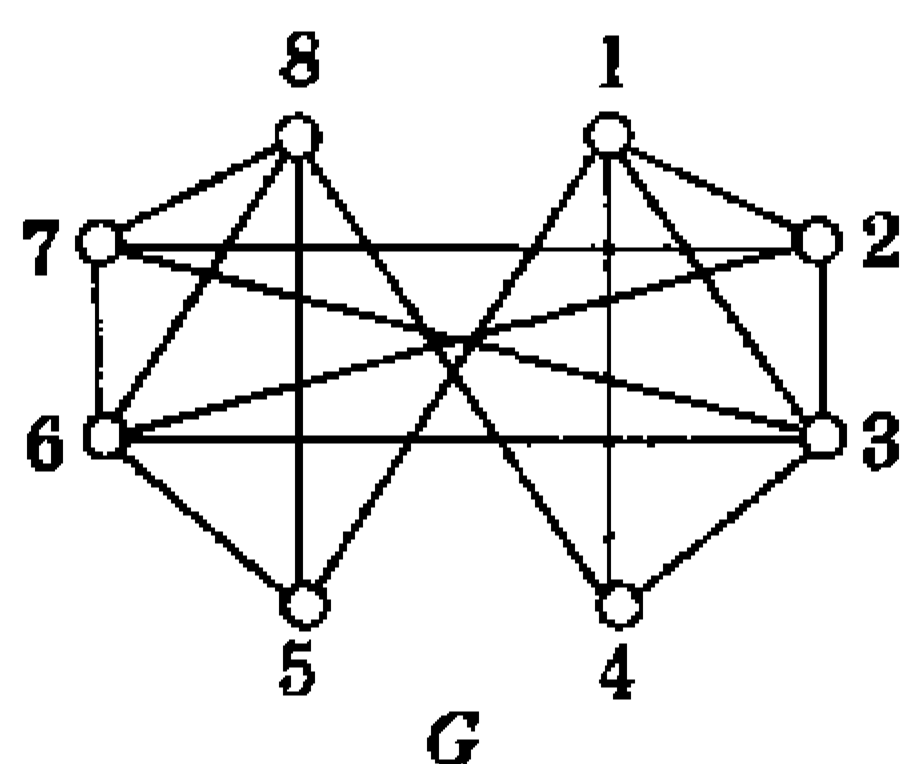
(7)若 $e>3v-6$ 或 $\delta>5$,则 G 是平面图.

如果不满足条件(6)、(7),G 的平面性需进一步检测.在发现 *DMP* 算法之后,Hopcroft 和 Tarjan 又最早给出了一个复杂性为 $O(n)$的算法,由于该算法的解释与证明所占篇幅太长,在此仅介绍 *DMP* 算法.

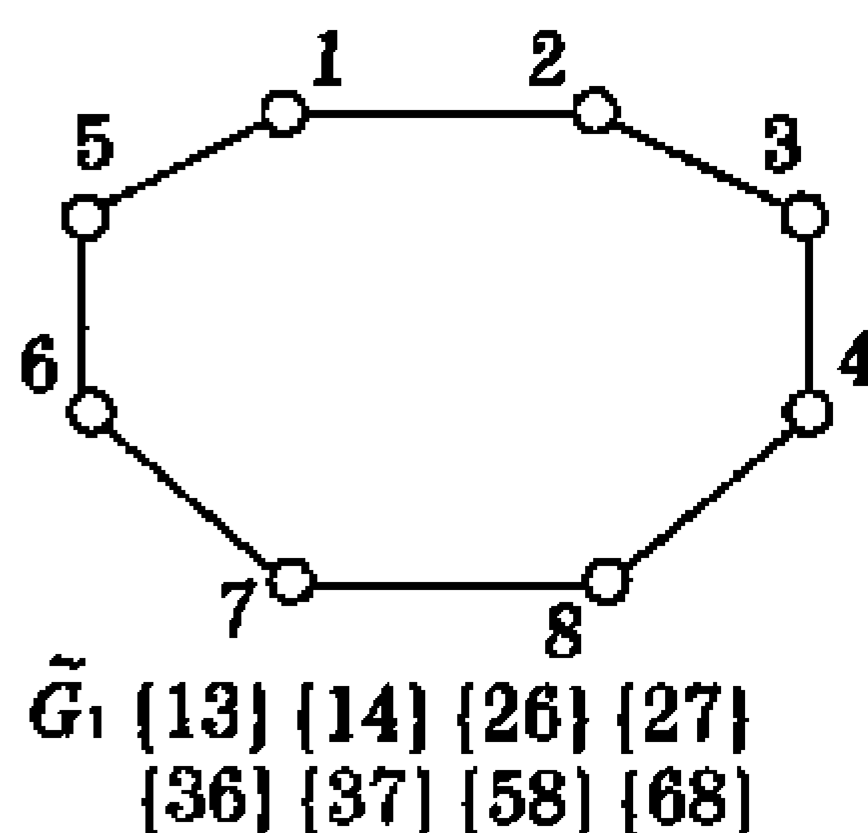
DMP 算法

(1)设 G_1 是 G 中的圈(因为无圈图必是平面图),求出 G_1 的平面表示 $\tilde{G}_1$. 令 $i=1$.

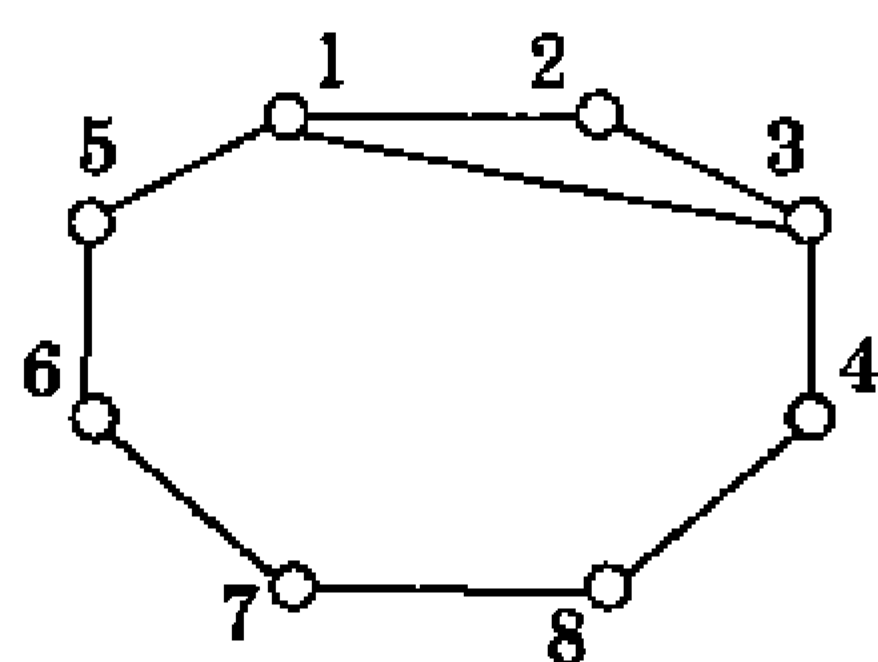
(2)若 $E(G)-E(G_i)=\emptyset$,则停止. 若 $E(G)-E(G_i)\neq\emptyset$,则确定 G 的所有 G_i 片,并对每个 G_i 片 B,求出集 $F_G(B,\tilde{G}_i)$.



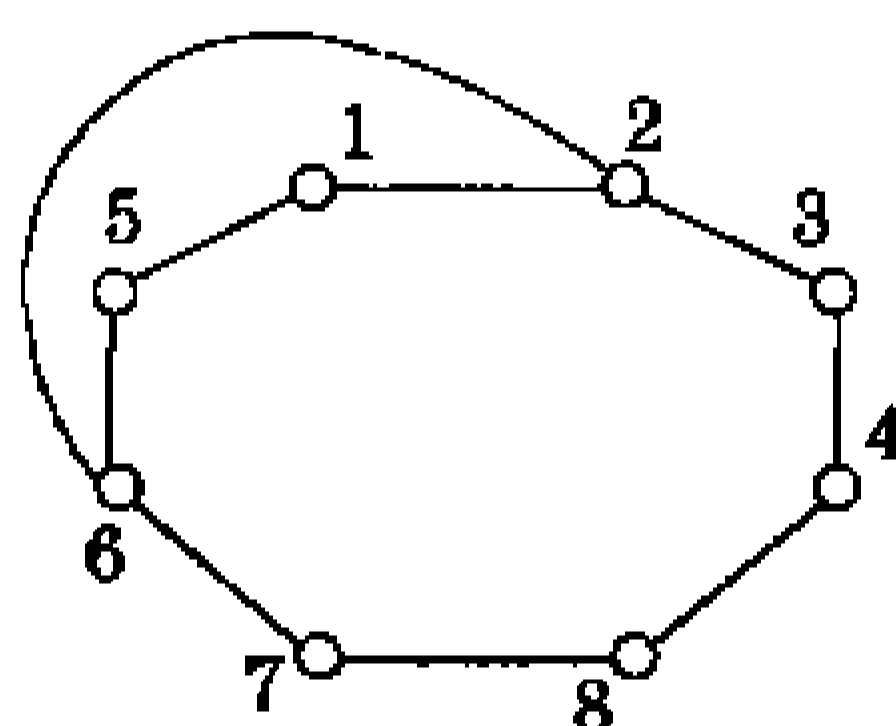
G



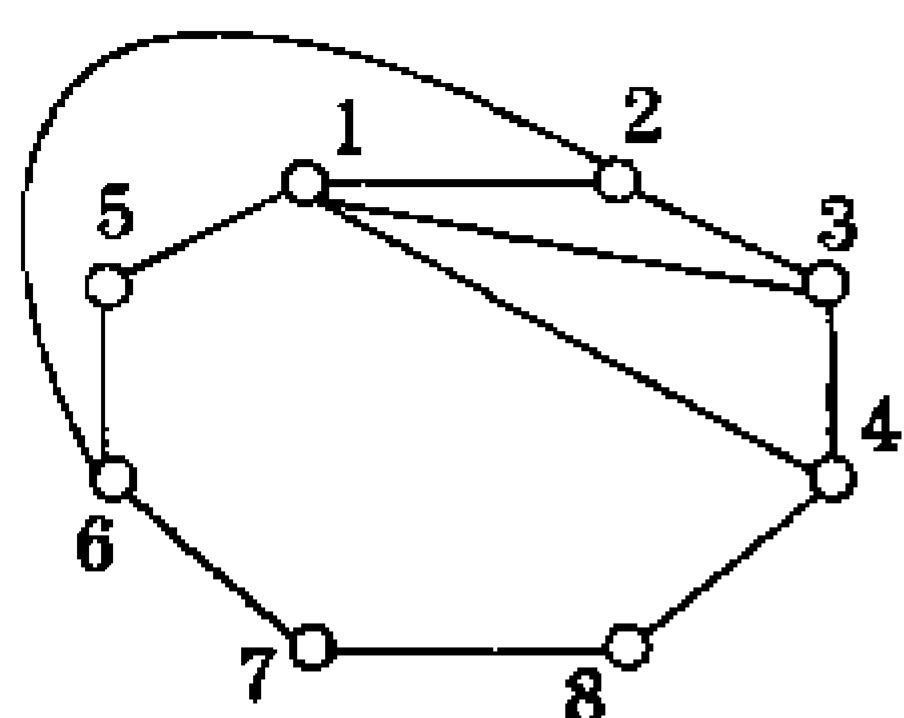
$\tilde{G}_1$ {13} {14} {26} {27} {36} {37} {58} {68}



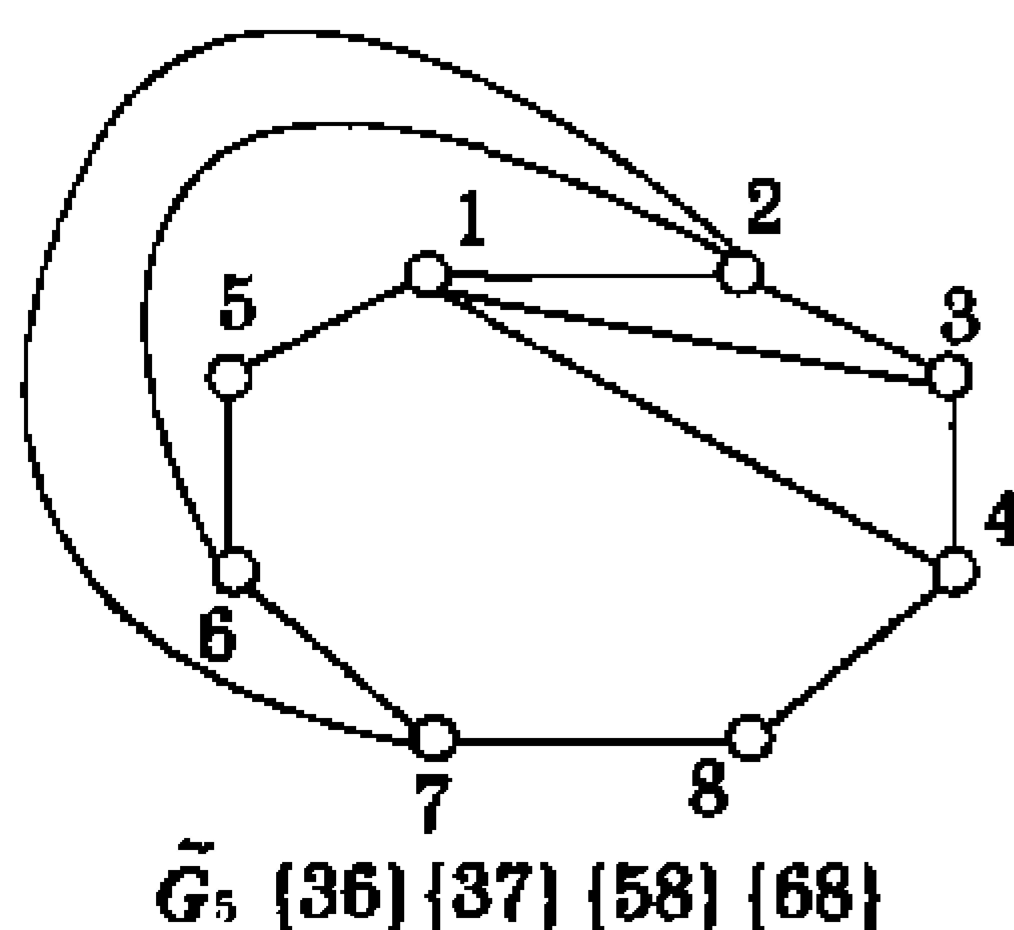
$\tilde{G}_2$ {14} {26} {27} {36} {37} {58} {68}



$\tilde{G}_3$ {14} {27} {36} {37} {58} {68}



$\tilde{G}_4$ {27} {36} {37} {58} {68}



$\tilde{G}_5$ {36} {37} {58} {68}

图 4-15

(3)若存在 G_i 片 B 使 $F_G(B,\tilde{G}_i)=\emptyset$,则停止,此时知 G 是非平面图.若存在 G_i 片 B 使 $|F_G(B,\tilde{G}_i)|=1$,则令 $\{f\}=F_G(B,\tilde{G}_i)$;若不存在这样的片 B,则令 B 是任何一个 G_i 片,并且任取 $f\in F_G(B,\tilde{G}_i)$.

(4)选取一条 xy 路 $P_i\subseteq B$ 使 $x,y\in V_G(B,G_i)$.令 $G_{i+1}=G_i\cup P_i$,并把 P_i 画在 $\tilde{G}_i$ 的面 f 内得 G_{i+1} 的一个平面表示 $\tilde{G}_{i+1}$.用 $i+1$ 代替之并转入第 2 步.

例 1 利用 *DMP* 算法检测图 4-15 所示图 G 的平面性.

解 从圈 $\tilde{G}_1=(123487651)$ 开始.以图 4-15 所示的顺序进行,可以看出作出图 $\tilde{G}_5$ 之后,存在 $\tilde{G}_5$ 片 $B=\{36\}$ 使 $F_G(B,\tilde{G}_5)=\emptyset$.算法停止,且 G 是非平面图.(为简便,片用其边集表示).

第四节 平面图的着色

定义 1 设 G 是无孤立结点的连通平面图,且 G 有 k 个面 $F_1,F_2,\cdots,F_k$(包括外部面).按下列过程作 G 的对偶图 G^*

(1)在 G 的每个面内设置一个结点 $v_i(1\leqslant i\leqslant k)$.

(2)过 F_i 与 F_j 的每一条公共边 e_k,作一条仅作一条边 $\{v_i,v_j\}(1\leqslant i,j\leqslant k)$ 与 e_k 相交.

(3)当且仅当 e_k 只是面 F_i 的边界时,v_i 恰有一自回路与 e_k 相交.

这样所得的图 G^* 称为图 G 的**对偶图(dual graph)**.若 G^* 与 G 同构,称 G 是**自对偶(self dual)**的.

如图 4-16 所示图 G,以虚线为边的图即为 G 的对偶图 G^*.

定义 2 图的**着色(coloring)**是对该图的每个顶点都指定一种颜色,使得没有两个相邻的顶点指定为相同的颜色.如果这些颜色选自于一个有 k 种颜色的集合而不管 k 种颜色是否都用到,这

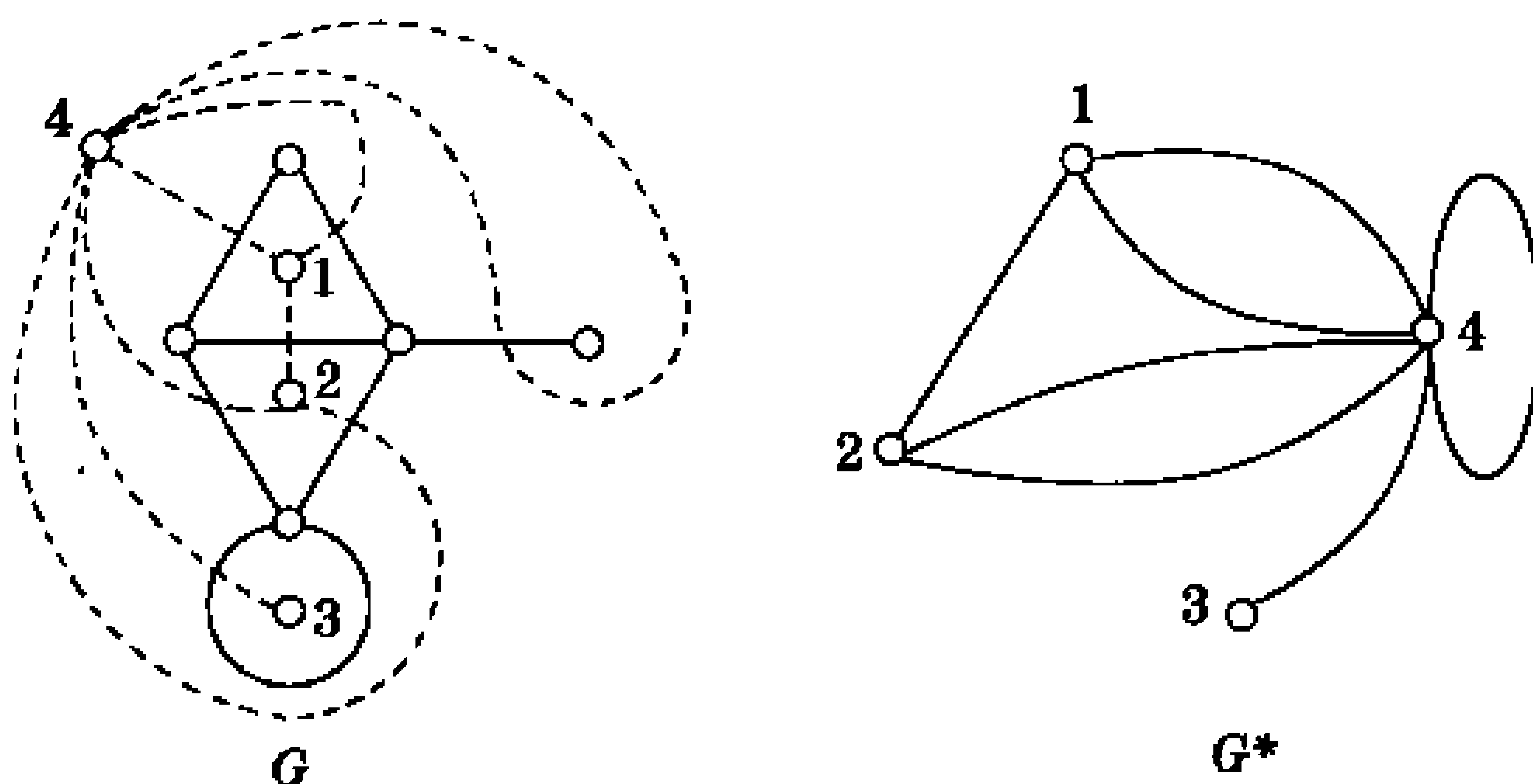


图 4-16

样的着色称为 k 着色.

图的着色又称为图的顶点着色(vertex coloring),k 着色也称为 k-顶点着色. 显然,考虑图的顶点着色,只需考虑简单图.

定义 3 图 G 的色数(chromatic number)是着色这个图 G 所需要的最少颜色数,记作 $x(G)$.

图 G 的色数也称为图 G 的点色数.

从定义不难发现,对于 G 的任何子图 H,均有 $x(H) \leqslant x(G)$. 若 G 是 n 阶完全图,则 $x(G)=n$;若 G 是至少有一边的二分图,则 $x(G)=2$;若 G 是长为奇数的圈,则 $x(G)=3$.

当 $x(G) \geqslant 3$ 时,G 的特征至今尚未清楚,在下一节,我们将给出 G 的色数 $x(G)$ 的一个上界.

定理 1 设 u 和 v 是图 G 中两个不相邻的顶点,则 $x(G)=\min\{x(G+\{u,v\}), x(G\cdot\{u,v\})\}$,其中 $G\cdot\{u,v\}$ 即把图 G 中结点 u 与 v 重合成一个新结点,且 G 中分别与结点 u 与 v 关联的边都与该新结点关联.

证明 记 $e=\{u,v\}$,设 $x(G)=k$,并考虑 G 的 k 着色. 假设顶点 u 与 v 染不同的颜色,则 G 的 k 着色也是 $G+e$ 的 k 着色. 因而,此时 $x(G+e) \leqslant k = x(G)$. 现假设顶点 u 和 v 的染色相同,

则 G 的一个 k 染色可得到 $G\cdot e$ 的一个 k 染色，于是，有 $x(G\cdot e) \leq k = x(G)$. 由于在 G 的 k 染色中，或者 u 与 v 染为不同的颜色或者为相同颜色，所以

$$\min\{x(G+e), x(G\cdot e)\} \leq x(G)$$

由于 G 是 $G+e$ 的子图，显然 $x(G) \leq x(G+e)$.

设 $x(G\cdot e) = k_1$，并把结点 u 与 v 重合所得的新结点记为 y，则在 $G\cdot e$ 的 k_1 着色中，把分配给 y 的颜色分配给 G 中 u 和 v，即可得到 G 的一个 k_1 着色. 于是 $x(G) \leq k_1 = x(G\cdot e)$，因此 $x(G) \leq \min\{x(G+e), x(G\cdot e)\}$.

综上所述，有

$$x(G) = \min\{x(G+\{u,v\}), x(G\cdot\{u,v\})\}.$$

平面地图实际上是一种平面图，它的面代表国家，边表示国家之间的边界，而点则是边界的交汇处. 对平面地图(假定地图里所有区域都是连通的)G 的着色，就是使地图边界相邻的国家有不同的颜色. 显然，对地图 G 的面着色问题等价于对 G 的对偶图 G^* 的顶点进行着色的问题.

求平面图的色数等于是求平面地图着色所需要的最少颜色数，使得没有两个相邻的区域指定为相同的颜色.

四色问题(four-colour problem) 连通简单平面图的色数不超过 4.

四色问题的起源有些模糊，但是可以肯定，在 1852 年，盖思里(Guthrie)把四色猜想转告了他的老师德·摩根(Augustus De Morgan)，德·摩根对这个问题极其感兴趣并且向数学界公布了它. 四色猜想曾一度被列为与数论中的 Fermat 猜想，函数论中的 Riemann 假设相提并论的三大数学难题之一，受到世界上许多最有才华的数学家的冲击. 历史上曾有许多人宣布证明了它，但都被后人一一否定. 1976 年，美国数学家肯尼思·阿佩尔(Kenneth Appel)和沃尔夫冈·黑肯(Wolfgang Haken)宣布了一个借助于计算机的证

明.但是,这个证明引起了广泛争论,原因是在他们的证明中使用了超过1000h的机时,它是不是真正的证明?

定理2(五色定理) 连通简单平面图 G 的色数为5.

证明 对图的顶点数作归纳.

当顶点数 $n\leqslant 5$ 时,显然成立.假设 $n-1$ 个顶点时成立,现证明 n 个顶点时也成立.

由于 G 是平面图,则 $\delta(G)\leqslant 5$.因此,G 中至少存在一顶点 v_0,其度数 $d(v_0)\leqslant 5$.在图 G 中删去顶点 v_0 得图 G',由归纳假设知 G' 的色数为5.然后将 v_0 又加回去,有两种情况:

(1)$d(v_0)<5$ 或 $d(v_0)=5$ 但和 v_0 邻接的5个结点着的颜色数小于5.则 v_0 极易着色,只要选与四周顶点不同的颜色着色即可.

(2)$d(v_0)=5$ 且和 v_0 邻接的5个结点着的是5种颜色,如图4-17(a)所示.为方便,我们称 G' 中所有红黄色顶点为红黄集,称 G' 中所有黑白色顶点为黑白集.于是又有两种可能.

①v_1 和 v_3 属于红黄集导出子图的两个不同块中,如图4-17(b)所示.将 v_1 所在块的红黄色对调,并不影响 G' 的正常着色.然后将 v_0 着上红色,即得图 G 的正常着色.

②v_1 和 v_3 属于红黄集导出子图的同一个块中,则 v_1 和 v_3 之间必有一条顶点属于红黄集的路 P,P 加上结点 v_0 可构成圈 C:$v_0v_1pv_3v_0$,如图4-17(c)所示.由于 C 的存在,将黑白集分为两个子集,一个在 C 内,另一个在 C 外,于是黑白集的导出子图至少有两个不同的块,一个在 C 内,一个在 C 外.于是问题转化为①的类型,对黑白集按①的办法处理,即得图 G 的正常着色.证毕.

例1 图4-18所示的图 G 和 H 的色数是多少?

解 图 G 的色数至少为3,因为顶点 a,b 和 c 必须指定为不同的颜色.为检验是否可以用三种颜色来对 G 着色,指定 a 为红

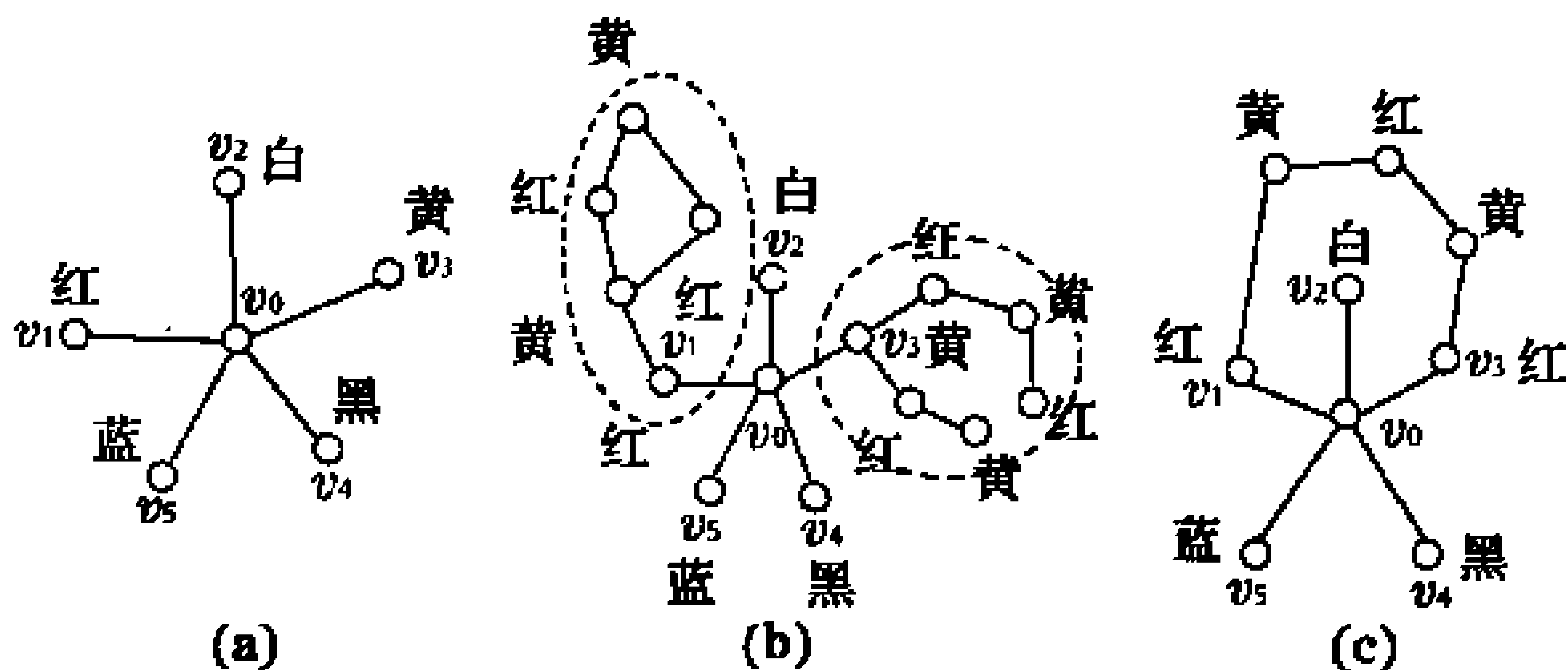


图 4-17

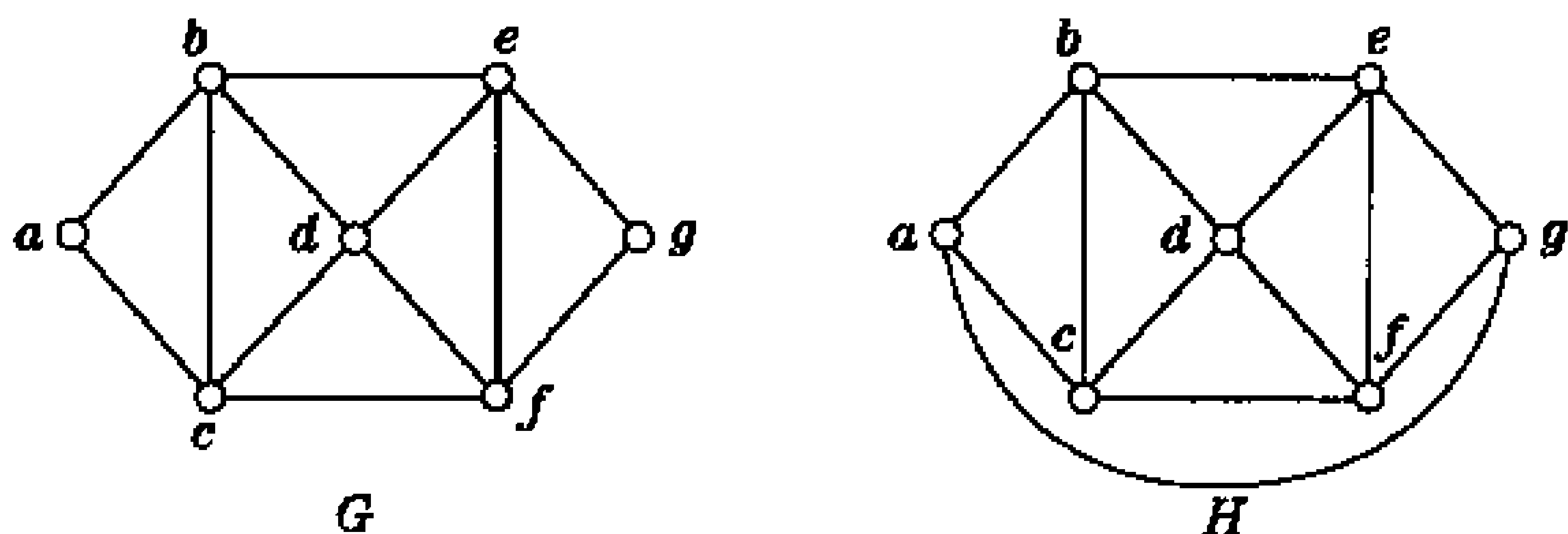


图 4-18

色，b 为蓝色，c 为绿色；从而 d 必为红色；e 为绿色，f 为蓝色；最后，确定 g 为红色. 因此 G 的色数 $\chi(G)=3$. H 的色数 $\chi(H)=4$.

例 2 由 $n(n\geqslant 3)$ 个顶点 $v_1, v_2, \cdots, v_n$ 以及边 $\{v_1, v_2\}$ $\{v_2, v_3\}\cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 组成的图称为圈图，记作 C_n. 试问圈图 C_n 的色数是多少?

解 当 n 是偶数时，对 C_n 着色需要两种颜色. 为构造这样的着色，任挑一个顶点把它着成红色，然后以顺时针方向绕图前进，把第二个顶点着成蓝色，把第三个顶点着成红色，依次类推. 第 n 个顶点着成蓝色，因为与它相邻的两个顶点(即第一个顶点和第 $n-1$ 个顶点)都着成红色.

当 n 是奇数时，C_n 的色数为 3. 为了看出这一点，挑选一个初始顶点. 为了只用两种颜色，当以顺时间方向遍历这个图时，必须使用交替的颜色. 而第 n 个顶点的两个相邻顶点（即第一个顶点和第 $n-1$ 个顶点）是同色，因此第 n 个顶点必须使用第三种颜色.

例 3 K_n 和 $K_{m,n}$ 的色数分别为多少？

解 由于 K_n 的每两个顶点都相邻，而两个相邻的顶点必须指定不同的颜色，因而 K_n 的色数为 n.

$K_{m,n}$ 的色数为 2. 用一种颜色着色 m 个顶点，用另一种颜色着色 n 个顶点.

已知的最好的求图的色数的算法（对图的顶点数来说）具有指数的最坏情形时间复杂性. 即使求图的色数的近似值的问题也是困难的. 已经证明，假如存在具有多项式最坏情形时间复杂性的算法，图的色数可以达到 2 倍近似值（即构造出一个不超过图的色数的两倍的界限），那么也存在具有多项式最坏情形时间复杂性的求图的色数的算法.

第五节 图着色的应用

图着色在图论中那么令人感兴趣，毫无疑问是由于它与四色问题有关. 同时，图着色有许多应用，又为图着色注入了活力. 本节我们介绍几个应用的例子.

贮藏问题（storage problem）某工厂生产 n 种化学制品 $c_1, c_2, \cdots, c_n$，其中某些制品是互不相容的. 若它们相互接触，则会引起爆炸. 作为一种预防措施，该工厂必须把仓库分成若干隔间，以便把不相容的化学制品贮藏在不同的隔间里，试问这个仓库至少应该分成几个隔间？

我们构作简单无向图 $G=\langle V,E\rangle$,其中

$$V(G)=\{C_1,C_2,\cdots,C_n\}$$

边$\{C_i,C_j\}\in E(G)\Leftrightarrow$化学制品 C_i 与C_j 互不相容.

于是,不难看出,仓库的最小隔间数等于图 G 的色数$x(G)$.

电视频道分配问题 某地区内有 n 家电视发射台 $T_1,T_2,\cdots,T_n$.主管部门为每家电视发射台分配一个频道.为排除干扰,使用同一频道的发射台之间的距离必须大于指定的正数 d.试问该地区至少需要多少频道:

构作简单无向图 $G=\langle V,E\rangle$,其中

$$V(G)=\{T_1,T_2,\cdots,T_n\}$$

边$\{T_i,T_j\}\in E(G)\Leftrightarrow T_i$ 与 T_j 之间距离$\leqslant d$.
于是需要的最小频道数等于 $x(G)$.

考试安排问题某高校有 n 门选修课程 $v_1,v_2,\cdots,v_n$ 需要进行期末考试.同一学生不能在同一天里参加两门课程的考试.问该校的期末考试至少要几天?

构作简单无向图 $G=\langle V,E\rangle$,其中

$$V(G)=\{v_1,v_2,\cdots,v_n\}$$

边$\{v_i,v_j\}\in E(G)\Leftrightarrow v_i$ 和 v_j 被同一位学生选修.
于是考试需要的最小天数等于 $x(G)$.

例如,需安排七门课期末考试.课程编号为 1 至 7.下列成对的课程有公共的学生选修:(1,2),(1,3),(1,4),(1,7),(2,3),(2,4),(2,5),(2,7),(3,4),(3,6),(3,7),(4,5),(4,6),(5,6),(5,7),(6,7).

构造如图 4-19 所示的图 G.显然,图 G 的色数$x(G)=4$.所以需要 4 个时间段.相应的安排如下:

时间数	课程
Ⅰ	2,6

Ⅱ	4,7
Ⅲ	3,5
Ⅳ	1

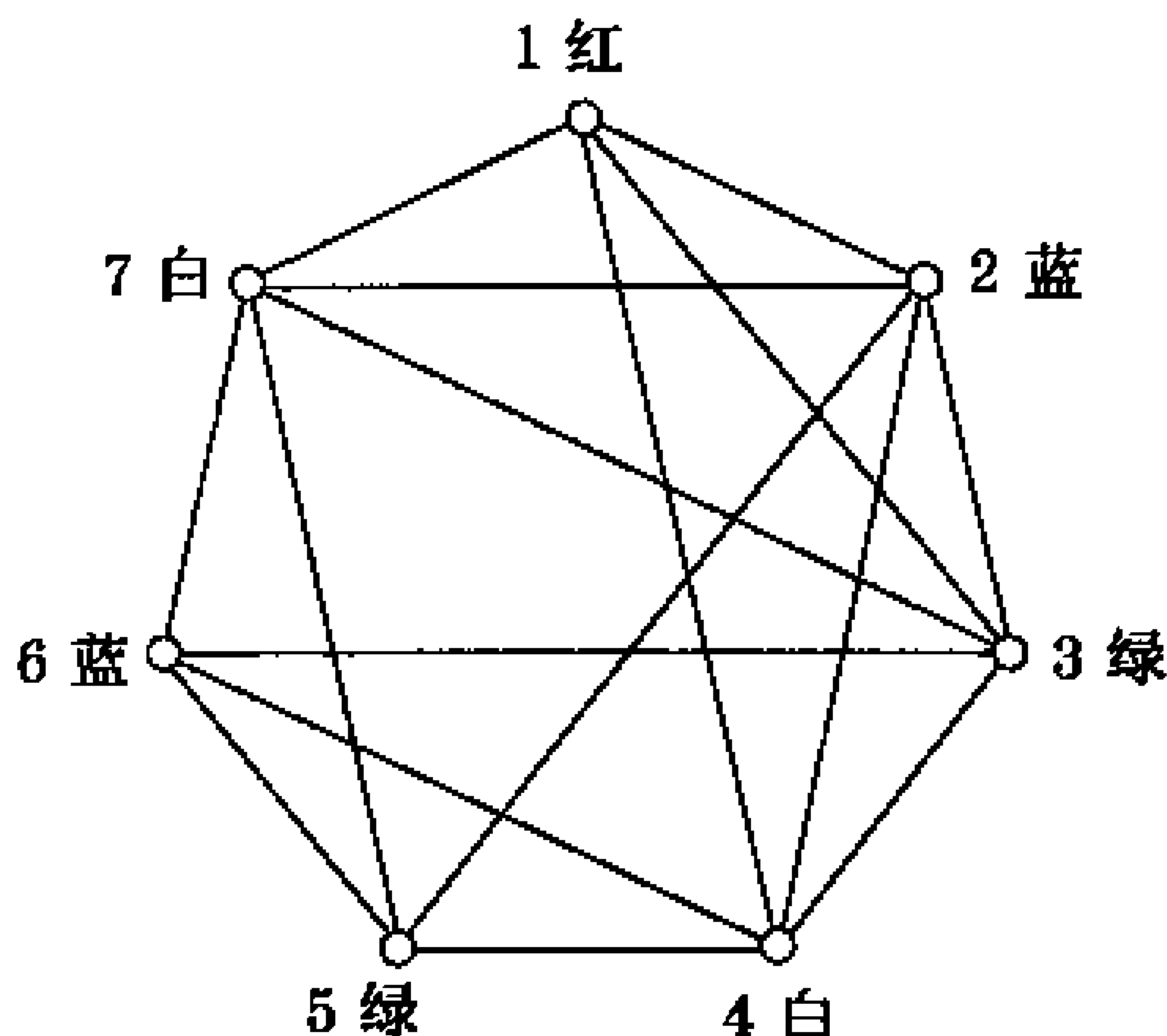


图 4-19

变址寄存器在有效的编译器里,当把频繁使用的变量暂时保存在中央处理单元而不是保存在常规内存时,可以加速循环的执行.对于给定的循环来说,需要多少个变址寄存器?可以用图着色模型来讨论这个问题.为了建立这个模型,设图的每个顶点表示循环里的一个变量.若在循环执行期间两个顶点所表示的变量必须同时保存在变址寄存器里,则在这两个顶点之间有边.因此,所需要的变址寄存器数就是这个图的色数.

遗憾的是,目前还没有一个有效算法来确定色数.下面介绍一个求 $\chi(G)$ 的近似有效算法——**顺序着色算法(sequential coloring algorithm)**.

设 $G=\langle V,E\rangle$ 是简单无向图,$V=\{x_1,x_2,\cdots,x_n\}$.为方便,用 $N(x_i)$ 表示与 x_i 相邻的全部结点的集合;对结点 x_i 着色 C,记作 $\pi(x_i)=C$.

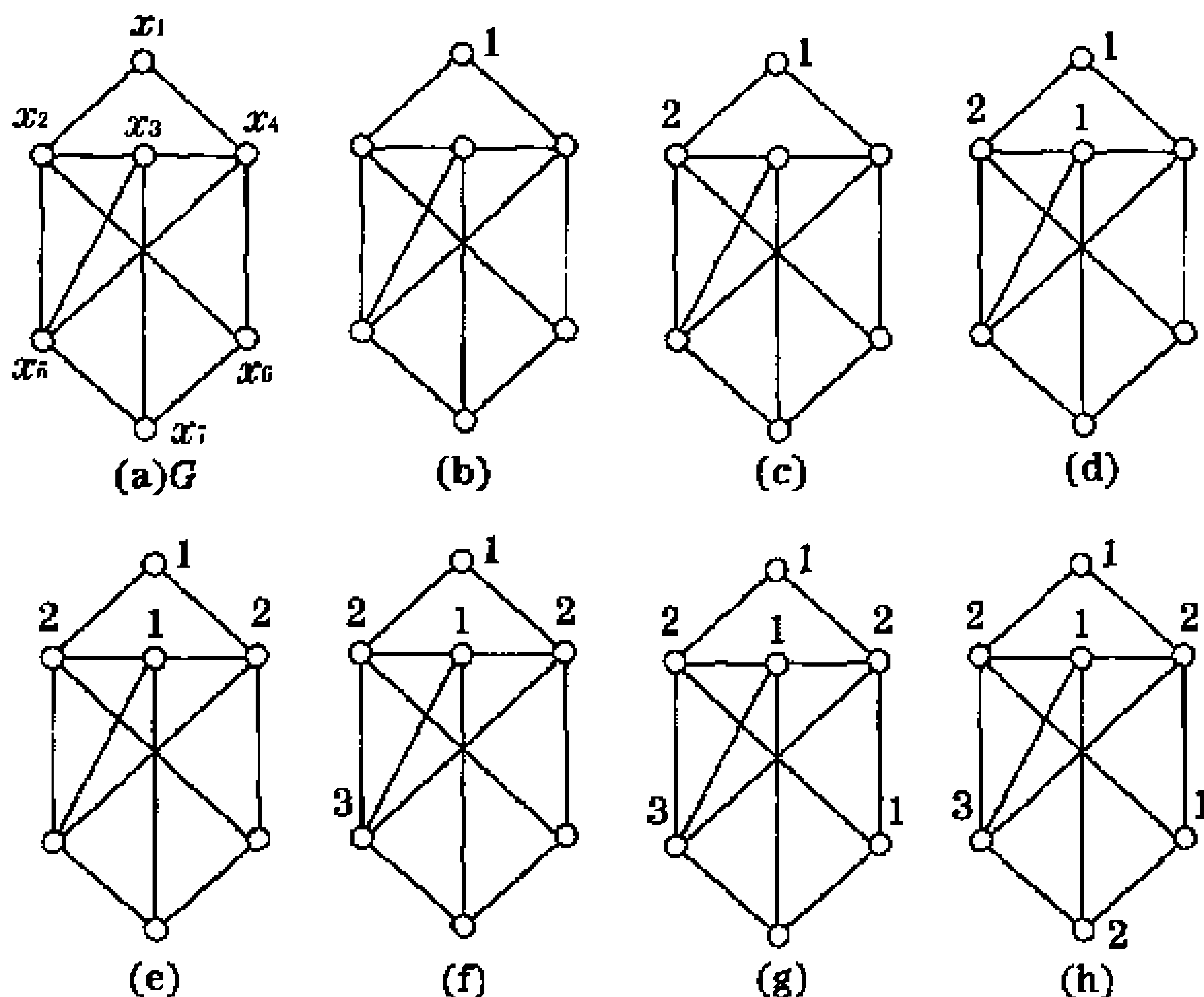


图 4-20

(1)$i := 1$

(2)$c := 1$

(3)若对 $\forall y \in N(x_i)$，$\pi(y) \neq c$，则令 $\pi(x_i) = c$ 并转入第(5)步.

(4)$c := c + 1$，并转第(3)步.

(5)若 $i < n$，则 $i := i + 1$，并转回第(2)步，否则停止.

例 1 试用顺序着色算法求图 4-20(a)所示图 G 的色数.

解 该算法执行的步骤如图 4-20(b)~(h)所示.

该算法与顶点标号顺序有密切的关系，这就是为什么叫顺序着色算法的原因. 例如，考虑二分图 $G = \langle V, E \rangle$，$V = Z \cup Y$，$Z = \{x_1, x_2, \cdots, x_n\}$，$Y = \{y_1, y_2, \cdots, y_m\}$，$E = \left\{ \{x_i, y_j\} \middle| i \neq j \right\}$.

若按 $x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_m$ 顺序执行该算法，则只需用最小颜色数目 2；若按 $x_1, y_1, x_2, y_2, \cdots, x_n, y_n$ 的顺序来执行该算法，则需要$\frac{1}{2}(n+m)$种颜色．如图 4-21(a)所示二分图 G，图 4-21(b)与(c)给出了按上述两种不同顺序进行着色．

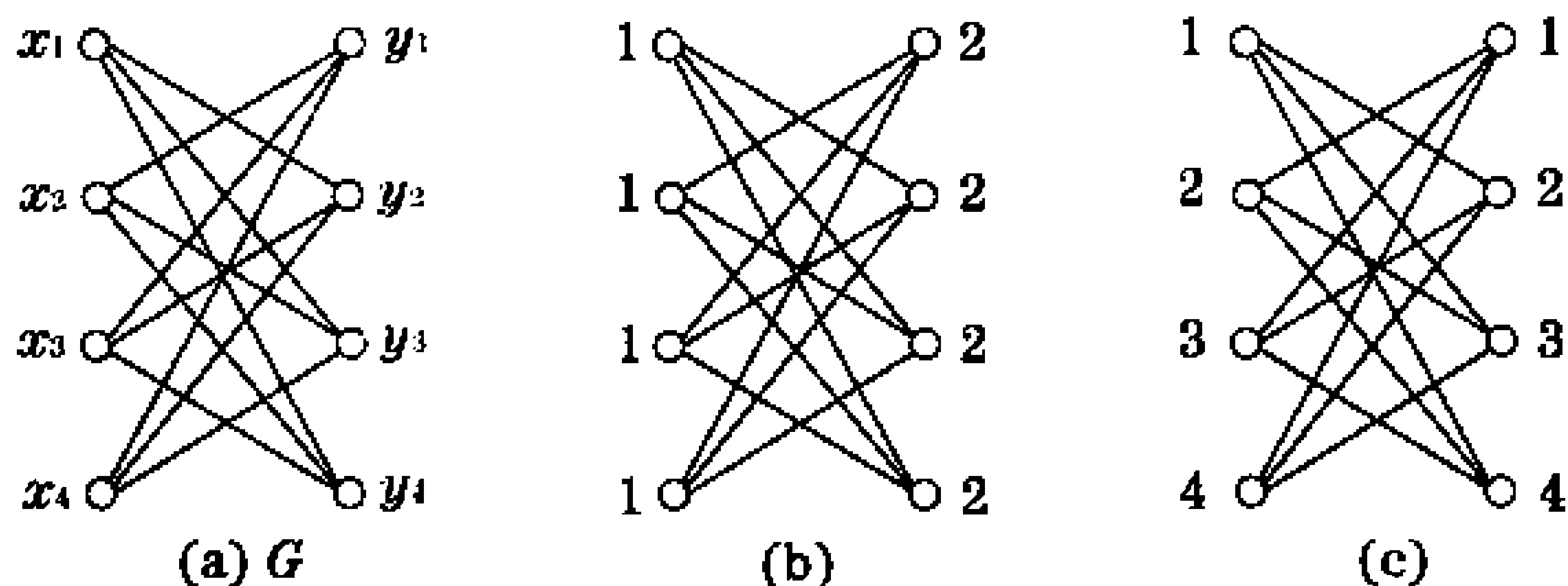


图 4-21

对 n 个结点的简单无向图 G，顺序着色算法的时间复杂度为 $O(n^2)$．设 C_0 是该近似算法求出的颜色数目，$c = x(G)$．由上述说明可知，该算法的性能比 c_0/c 是任意大．目前还没有找到一个有效近似的着色算法，其性能比为不超过一个常数．

定理 1 设 G 是连通简单图，图 G 的最大顶点度为 Δ，那么顺序着色算法产生 G 的顶点的一个 $\Delta+1$——着色，因此 $x(G) \leqslant \Delta + 1$．

证明 顺序着色算法用语言描述就是依次考虑每一个顶点，并将尚未指定给与其邻接的顶点的最小颜色指定给该顶点，特别是决不能将两个邻接顶点指定为相同的颜色，因此顺序着色算法确实产生一个顶点着色．最多存在 Δ 个顶点与顶点 x_i 邻接，于是，在 $x_1, x_2, \cdots, x_{i-1}$ 中最多有 Δ 个与 x_i 邻接．所以，当算法进行到对顶点 x_i 着色时，在颜色 $1, 2, \cdots, \Delta+1$ 中至少有一种颜色尚未指定给与 x_i 邻接的顶点，并且算法将这些颜色中最小的指定给

x_i. 于是顺序着色算法产生图 G 的顶点的一个 $\Delta+1$——着色. 证毕

定理 1 给出了连通简单图 G 的色数的上界. 1941 年, R. L. Brooks 证明了下面的定理.

定理 2 设 G 是一个连通简单图, 其顶点的最大度为 Δ. 如果 G 既不是完全图 K_n, 也不是奇数阶圈图 C_n, 则 $x(G)\leqslant\Delta$.

第六节 边着色

定义 1 图 G 的**边着色**是对 G 的每条边都指定一个颜色, 使得没有两个相邻的边都指定为同一种颜色. 如果这些颜色都取自一个有 K 种颜色的集合而不管 K 种颜色是否都用掉, 这样的边着色称为 K——**边着色**.

定义 2 图 G 的**边色数(edge chromatic number)**是着色这个图 G 所需要的最少颜色数, 记作 $x'(G)$.

定理 1 如果 G 是非空连通简单图, G 的最大顶点度为 Δ, 则 $\Delta\leqslant x'(G)\leqslant\Delta+1$.

证明 定理的下界显然成立. 关于上界, 对 $|E(G)|=q$ 进行归纳证明: 任意连通简单图 G 存在 $\Delta(G)+1$——边着色. 当 $q=1$ 时, 这个结论显然成立. 假设这个结论对边数小于 q 的所有图成立, 这里 $q\geqslant 2$. 设 G 是有 q 条边的图, 我证明 G 是 $\Delta(G)+1$——边着色的.

设 $e=\{u,v\}$ 是 G 的边, 根据归纳假设, $G-e$ 是 $\Delta(G-e)+1$——边着色的. 由于 $\Delta(G-e)\leqslant\Delta(G)$, 所以 $G-e$ 也是 $\Delta+1$——边着色的.

设已给 $G-e$ 的 $\Delta(G)+1$——边着色, 即除 e 外 G 的每条边已经由 $\Delta(G)+1$ 种颜色的某种着色使得邻接边着不同颜色. 在 G 的这个局部 $\Delta(G)+1$——边着色中, 对任意顶点 $x\in V(G)$,

如果与 x 关联的所有已经染色边的颜色都不同于 a，则称颜色 a 在顶点 x 上未出现，否则，就称颜色 a 在顶点 x 上出现. 对图 G 的任何顶点 x，由于 $\deg(x)\leqslant\Delta(G)$，故至少有一种颜色在顶点 x 上未出现.

设 $e=e_0$，$v=v_0$，并且颜色 a_1 在 v_0 上未出现. 如果颜色 a_1 在顶点 u 上也未出现，则可用颜色 a_1 染边 $\{u,v\}$，从而得到 G 的 $\Delta(G)+1$——边着色，定理成立. 从而颜色 a_1 在顶点 u 上出现，即存在与 u 关联的边 $e_1=\{u,v_1\}$ 染颜色 a_1. 设颜色 a_2 在顶点 v_1 上未出现，则 $a_2\neq a_1$. 如果 a_2 在顶点 u 上也未出现，则将边 $\{u,v_1\}$ 重新染为颜色 a_2，而将边 $\{u,v_0\}$ 染为颜色 a_1，即可得到 G 的 $\Delta(G)+1$——边着色，故颜色 a_2 在顶点 u 上也出现. 假设已经得到一组边序列 $\{u,v_0\}$，$\{u,v_1\}$，…，$\{u,v_{k-1}\}$ $(k>2)$ 以及颜色序列 a_1，a_2，…，a_k 满足：

(1)对任意 $1\leqslant i\leqslant k-1$，边 $\{u,v_i\}$ 染颜色 a_i；

(2)对任意 $1\leqslant i\leqslant k$，颜色 a_i 在顶点 v_{i-1} 上未出现；

(3)对任意 $1\leqslant i<j\leqslant k$，$a_i\neq a_j$；

(4) a_k 在顶点 u 上出现.

此时，存在边 $\{u,v_k\}$ 染颜色 a_k 以及在顶点 v_k 上未出现的颜色 a_{k+1}. 下面分几种情况考虑.

情况 1　颜色 a_{k+1} 在顶点 u 上未出现. 对任意 $0\leqslant i\leqslant k$，将边 $\{u,v_i\}$ 染颜色 a_{i+1}，从而得到 G 的 $\Delta(G)+1$——边着色.

情况 2　存在 $1\leqslant j\leqslant k$ 使得颜色 $a_{k+1}=a_j$. 由于 $\{u,v_k\}$ 染颜色 a_k 且颜色 a_{k+1} 不在 v_k 上出现，所以 $j<k$. 设 a 为在顶点 u 上未出现的颜色. 如果颜色 a 在顶点 v_k 上也未出现，则将边 $\{u,v_k\}$ 着颜色 a，对任意 $0\leqslant i<k$，将边 $\{u,v_i\}$ 染颜色 a_{i+1}，从而得到 G 的 $\Delta(G)+1$——边着色. 类似可得当 a 在顶点 v_{j-1} 上未出

现时，存在 G 的 $\Delta(G)+1$——边着色，从而可假设颜色 a 在顶点 v_k 和 v_{j-1} 上均出现. 令 $M=\{e\mid e\in E(G)$，边 e 染颜色 $a\}$，$M'=\{e\mid e\in E(G)$，边 e 染颜色 $a_j\}$，且 $H=M\cup M'$. 因此，H 的每个分支为偶圈或者路. 又因为颜色 a 在顶点 u 上未出现且 $a_{k+1}=a_j$ 在顶点 v_k 和 v_{j-1} 均未出现，故 $d_H(u)=d_H(v_k)=d_H(v_{j-1})=1$，所以 u，v_k 和 v_{j-1} 不在 H 的同一个分支上，设 H 中不包含顶点 u 且包含顶点 v_k 或 v_{j-1} 的分支为路 P.

如果 $v_{j-1}\in V(P)$，则交换 P 上所有边的颜色使得 a 在顶点 v_{j-1} 和 u 上均未出现，用颜色 a 染边 $\{u,v_{j-1}\}$，对任意 $0\leqslant i\leqslant j-2$，用颜色 a_{i+1} 染边 $\{u,v_i\}$ 可得 G 的 $\Delta(G)+1$——边着色.

如果 $v_{j-1}\notin V(P)$，则 $v_k\in V(P)$. 因为 H 中包含 u 的分支同时包含顶点 v_j 且与 P 是不交的，所以变换 P 上所有边的颜色不会改变边 $\{u,v_j\}$ 的颜色，从而可类似得到 G 的 $\Delta(G)+1$——边着色.

情况 3　对任意 $1\leqslant i\leqslant k$，$a_i\neq a_{k+1}$. 此时，将边 $\{u,v_k\}$ 加入边序列，将颜色 a_{k+1} 加入颜色序列，则新的边序列和颜色序列满足条件(1)~(4). 由于颜色集为有限集，从而总会出现情况 1 或情况 2，故 G 是 $\Delta(G)+1$——边可染色的. 证毕.

根据定理 1，全部非空图集实际上能分成两类，如果 $x'(G)=\Delta(G)$，则称 G 为**第一类图**，否则称为**第二类图**，这样主要问题是确定所给图是第一类图还是第二类图.

例如，对于圈图 $C_n(n\geqslant 3)$，如果 n 是偶数，则 C_n 是第一类图，否则为第二类图.

定理 2　任意二分图 G 是第一类图.

证明　对二分图 G 的边数 q 进行归纳证明：存在 G 的 $\Delta(G)$——边染色. 当 $q=1$ 时，定理成立. 假定对所有边数小于 q $(q\geqslant 2)$ 的二分图，定理成立. 现设 G 是有 q 条边的二分图且

$e=\{u,v\}$是 G 的边.根据归纳假设,图 $G-e$ 存在$\Delta(G-e)$——边着色.由于 $d_{G-e}(u)<\Delta(G)$且 $d_{G-e}(v)<\Delta(G)$,故在顶点 u 和 v 上至少各有一种颜色未出现,如果颜色在顶点 u 和 v 上都未出现,则用颜色 a 染边 e 得到 G 的 $\Delta(G)$——边染色.如果颜色 a_1 在顶点 u 上未出现但在顶点 v 上出现,而颜色 a_2 在顶点 v 上未出现而在顶点 u 上出现.考虑 $G-e$ 中由所有 a_1 色边和所有 a_2 色边导出的子图 G',因此,G'的连通分支必为路或偶圈,而且顶点 u 和 v 都是路的端点.又因为顶点 u 和 v 分别属于二分图的不同部分集,所以 G'中的任意 uv 路为奇长路,但是在 G'中,顶点 u 和 v 上出现的颜色不同,所以顶点 u 和 v 一定不属于同一条路,即顶点 u 和 v 属于 G'的不同分支.这时只要把顶点 u 所在的分支上的所有边的颜色 a_1 和 a_2 对调使得在顶点 u 上颜色 a_2 也不出现,把边 e 染颜色 a_2 得 G 的 $\Delta(G)$——边染色.证毕.

下面再来考虑另一个问题:已知一个图 G 和 k 种颜色的集合 $\{1,2,\cdots,k\}$,那么存在图 G 的多少 k——着色?

定义 3 图 G 的不同 k——着色的数目,称为图 G 的**色多项式(chromatic polynomeals)**,记作 $f(G,k)$.

显然,若 $k<\chi(G)$,则 $f(G,k)=0$,而 $f(G,k)>0$ 的最小的 k 是 G 的色数.

设 m_i 是 i 种颜色对图 G 的顶点进行着色的不同方案数.用 k $(k\geqslant i)$种颜色对图 G 进行着色,每取 i 种颜色时,共有 $m_iC_k^i$ 种不同的方案,于是我们有

$f(G,k)=m_1C_k^1+m_2C_k^2+\cdots+m_nC_k^n$,其中 n 为图 G 的顶点数.显然,$f(G,k)$是 k 的多项式.

设 u,v 是图 G 的不邻接的两个顶点,把 u,v 收缩成一个顶点 x,并把 G 中凡是与 u,v 关联的边均使之与 x 关联,这样所得的新图记作 $\boldsymbol{G\cdot uv}$.把图 G 的一条边 e 删去并使它的两个端点重

合，也即边 e 被收缩，这样得到的新图记作 $G:e$. 如图 4-22 所示.

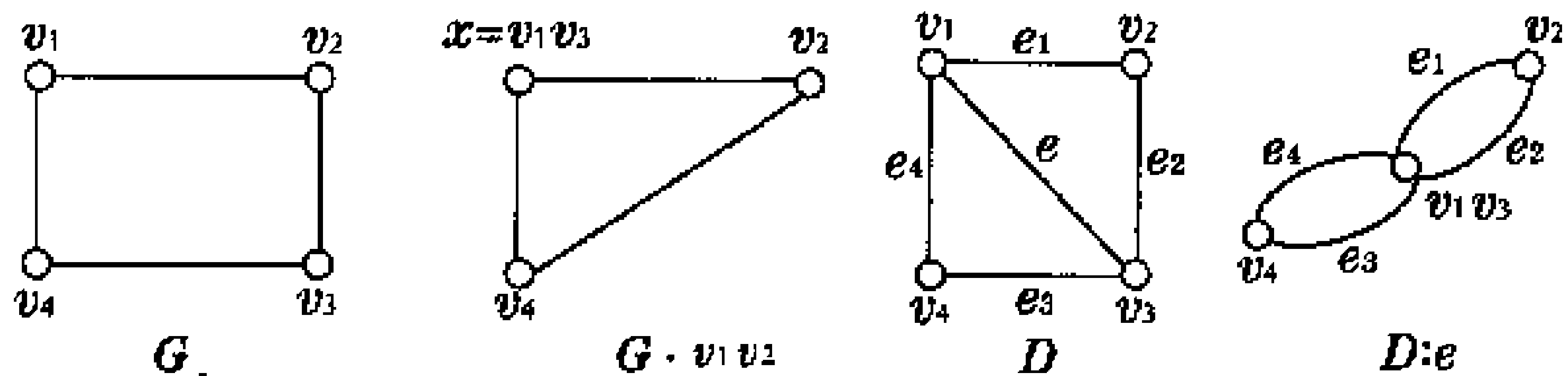


图 4-22

定理 3 设 u,v 是图 G 中不邻接的两个顶点，则 $f(G,k)=f(G+\{u,v\},k)+f(G\cdot uv,k)$.

证明略.

推论 设 $e=\{u,v\}$ 是图 G 的边，则

$$f(G,k)=f(G-e,k)-f(G:e,k)$$

定理 3 表明，如果 G 是有 P 个顶点和 q 条边的图，则有一个 $q+1$ 条边的图 $G_1=G+\{u,v\}$（u 与 v 不邻接）和一个有 $P-1$ 个顶点的图 $G_2=G\cdot uv$，使 $f(G,k)=f(G_1,k)+f(G_2,k)$. 对 G_1 和 G_2 依次类推，直到只出现完全图为止. 因此，一个图的色多项式 $f(G,k)$ 是 $f(K_n,k)$ 型的表达式的和.

例 1 三阶完全图 K_3，有 $f(K_3,k)=k(k-1)(k-2)$.

这是因为，对 K_3 中的任何一个指定的顶点，可以用 k 种颜色中的任何一种进行着色；对 K_3 的第二个顶点，可以用 $k-1$ 种颜色中的任何一种着色；

第三个顶点则用 $k-2$ 种颜色中的任何一种着色.

一般地，对 P 阶完全图 K_p，有 $f(K_p,k)=k(k-1)(k-2)\cdots(k-p+1)$.

例 2 求图 4-23 所示图 G 的色多项式.

解 连续对图 G 进行加边和收缩两个过程，直到出现完全图为止. 如图 4-24 所示.

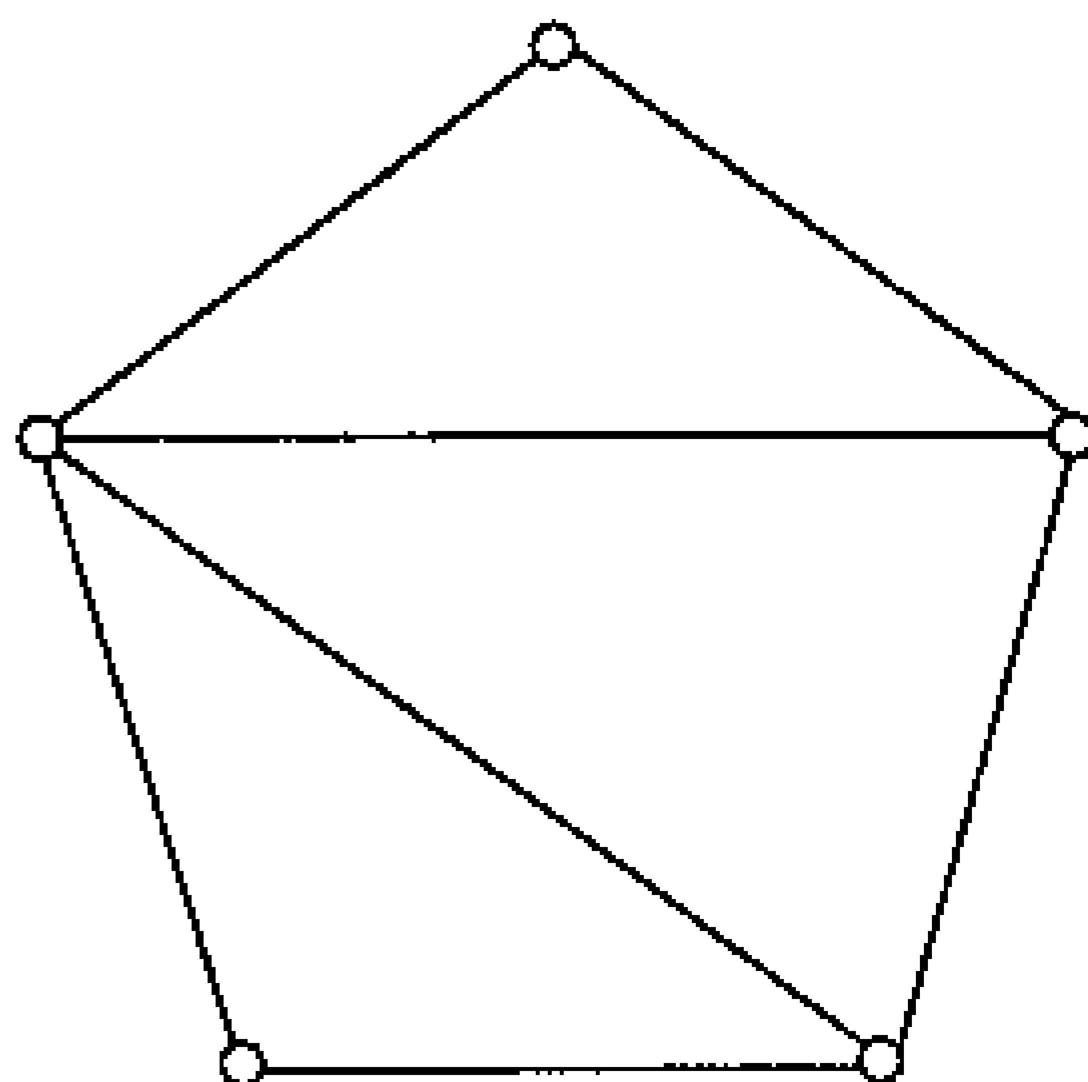


图 4-23

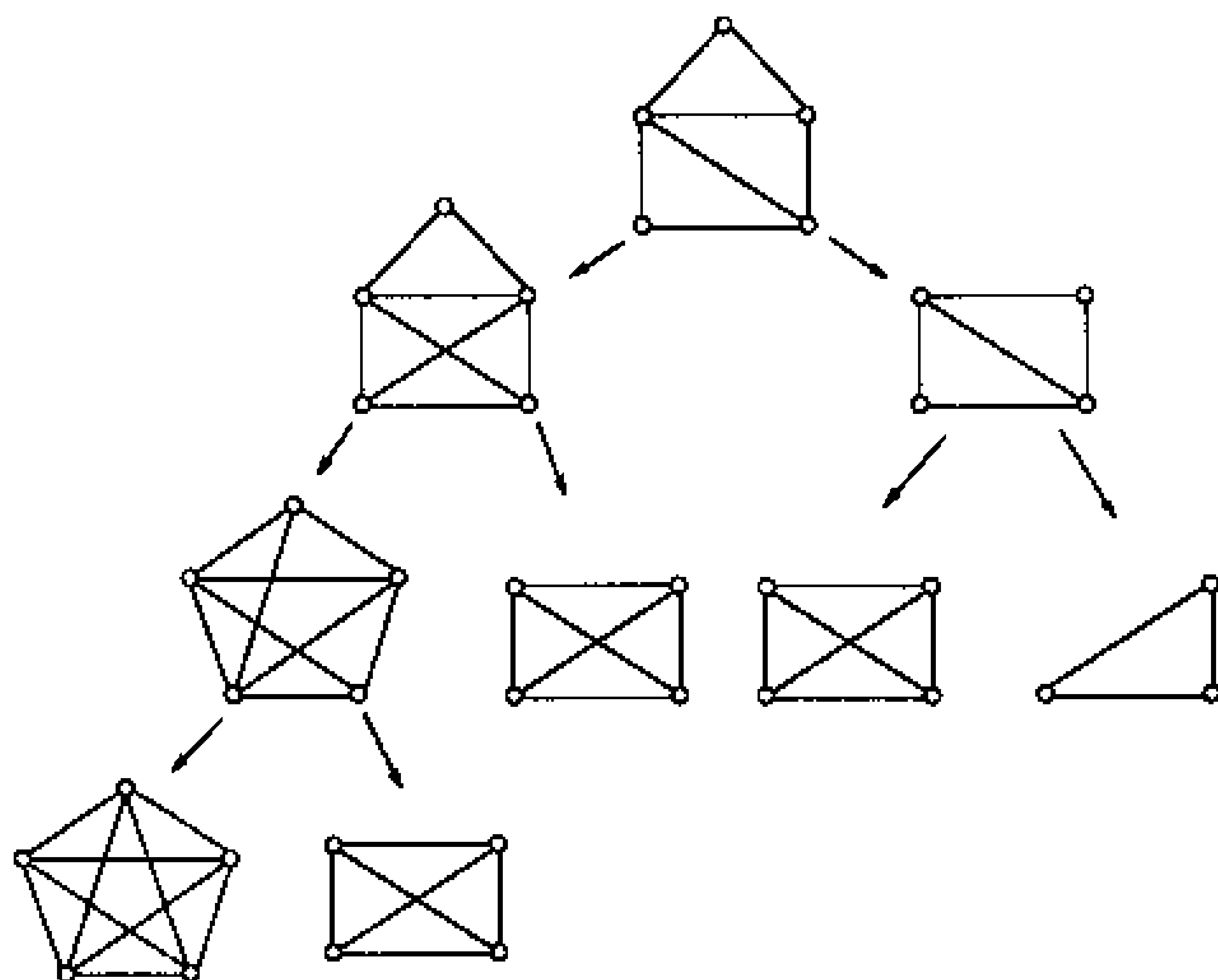


图 4-24

于是,有

$$f(G,k)=K_5+3K_4+K_3$$
$$=k(k-1)(k-2)(k-3)(k-4)+3k(k-1)(k-2)(k-3)+k(k-1)(k-2)$$

$$= k^5 - 7k^4 + 18k^3 - 20k^2 + 8k$$

定理 4　n 阶图 G 的色多项式 $f(G,k)$，其首项为 k^n，常数项为零，此外系数的符号为正负相间.

证明　对边数 ϵ 进行归纳证明. 显然，当 $\epsilon=0$ 时，定理成立. 这是因为，n 个顶点的空图的色多项式为 k^n.

假设定理对少于 ϵ 条边的图成立.

设 e 是 G 的一条边，则 $G-e$ 是有 n 个顶点 $\epsilon-1$ 条边的图，$G\cdot uv$ 是有 $n-1$ 个顶点和至少有 $\epsilon-1$ 条边的图. 根据归纳法假设，存在非负整数系数 $a_1,a_2,\cdots,a_{n-1}$ 和 $b_1,b_2,\cdots,b_{n-2}$ 使

$$f(G-e,k) = k^n - a_{n-1}k^{n-1} + a_{n-2}k^{n-2} - \cdots + (-1)^{n-1}a_1k$$

和 $f(G\cdot uv,k) = k^{n-1} - b_{n-2}k^{n-2} + b_{n-3}k^{n-3} - \cdots + (-1)^{n-2}b_1k$，由推论，有 $f(G,k) = f(G-\{u,v\},k) - f(G\cdot uv,k) = k^n - (a_{n-1}+1)k^{n-1} + \sum_{i=2}^{n-1}(-1)^i(a_{n-i}+b_{n-i})k^{n-i}$.

习　题　四

1. 假定一个连通平面图有 6 个顶点，每个顶点的度都为 4，问这个平面图有多少个面？

2. 假定一个连通平面图有 30 条边和 20 个面，问这个图有多

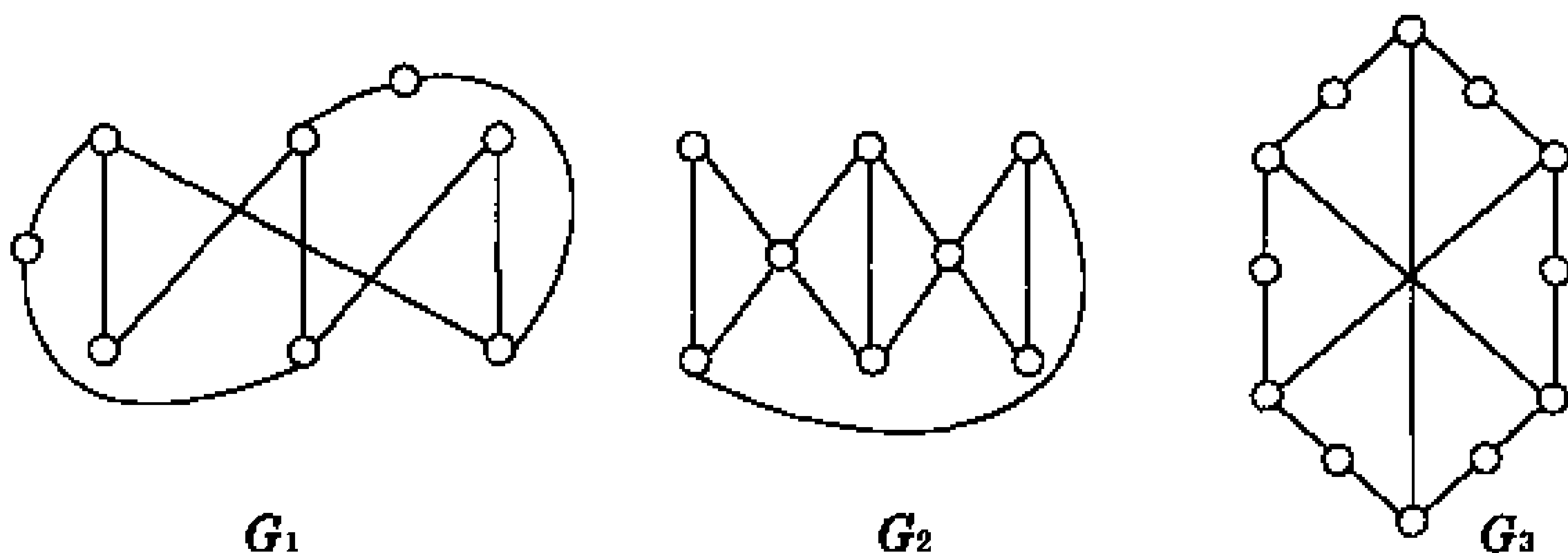


图 4-25

少个顶点?

3．确定图 4-25 所示的图 G_1、G_2 和 G_3 是否同胚于 $K_{3,3}$?

4．用库拉图斯基定理来确定图 4-26 所示的图 G_1、G_2 和 G_3 是否为平面图?

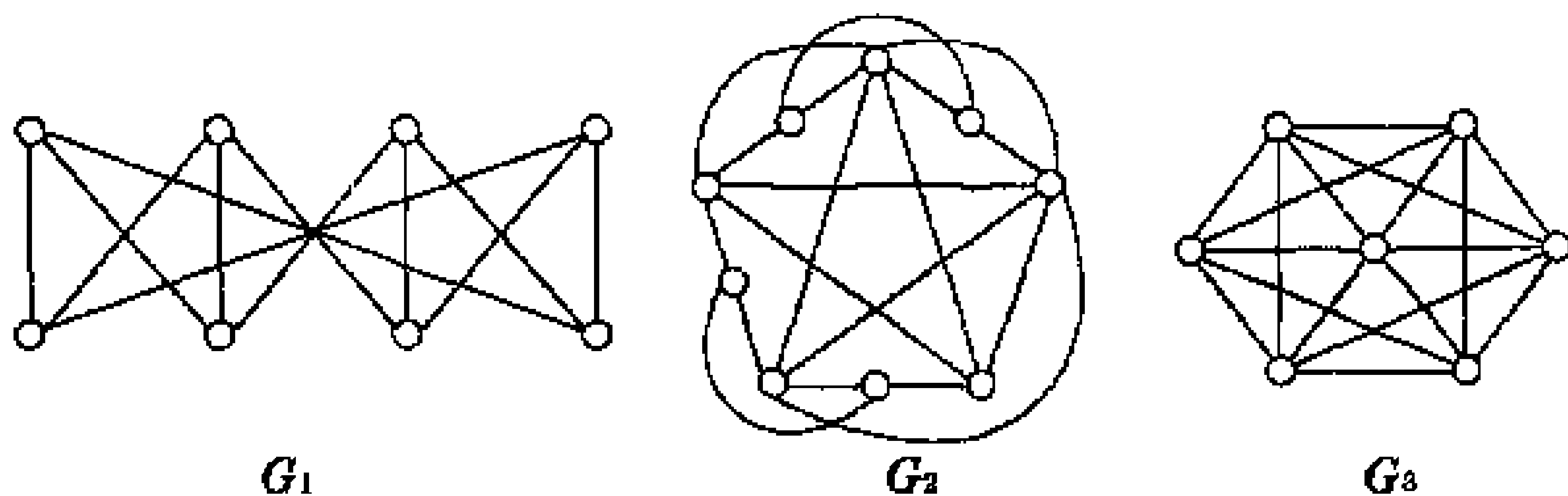


图 4-26

5．证明图 4-27 中所示的四个图均为平面图.

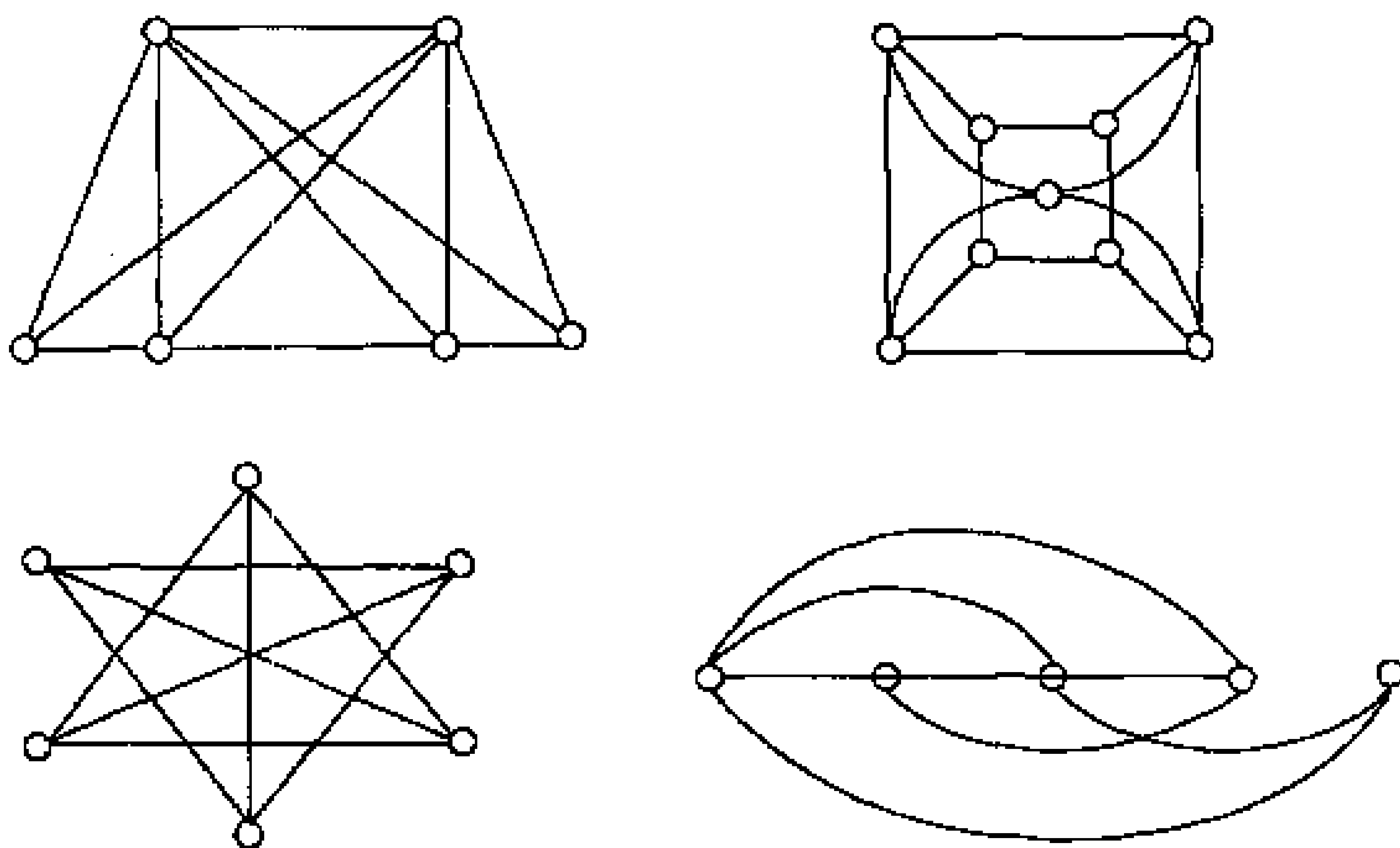


图 4-27

6．证明:极大平面图 G 一定是连通图.

7．证明:在顶点数 $v \geqslant 4$ 的极大平面图 G 中,G 中最小度

$\delta \geqslant 3$.

8. 如果 G 是 $v(v \geqslant 3)$ 个顶点 ε 条边的极大平面图，则 $\varepsilon = 3v - 6$.

9. 设简单平面图 G 中顶点数 $v = 7$，边数 $\varepsilon = 15$，证明 G 是连通的.

10. 简单图 G 的**厚度** $\boldsymbol{\theta(G)}$ 是指 G 的平面性子图的最小数，这些子图以 G 作为它们的并集. 设 G 是 $v(v \geqslant 3)$ 个顶点 ε 条边的简单图，证明 G 的厚度 $\theta(G)$ 满足：

(1) $\theta(G) \geqslant \left\{ \frac{\varepsilon}{3v-6} \right\}$；

(2) $\theta(G) \geqslant \left[\frac{\varepsilon + 3v - 7}{3v-6} \right]$；

其中，$\{x\}$ 为不小于 x 的最小整数，$[x]$ 为不大于 x 的最大整数.

11. 求下面各图的厚度：

1) K_5

2) 彼得森图；

3) 4 维立方体 Q_4；

4) $K_{3,3}$

12. 一个简单图的**交叉数**是指，当在平面里画出这个图，其中不允许任何三条边在同一个点交叉时，交叉的最少次数. 求下面每个非平面图的交叉数.

(1) K_5　(2) K_6　(3) K_7　(4)　$K_{3,3}$

(5) $K_{3,4}$　(6) $K_{4,4}$　(7) $K_{5,5}$

13. 对图 4-28 所示图 G 进行预处理，并验证其平面性.

14. 运 DMP 平面性算法证图 4-29 所示图 G_1 与 G_2 的平面性.

15. 求图 4-30 所示各图的对偶图.

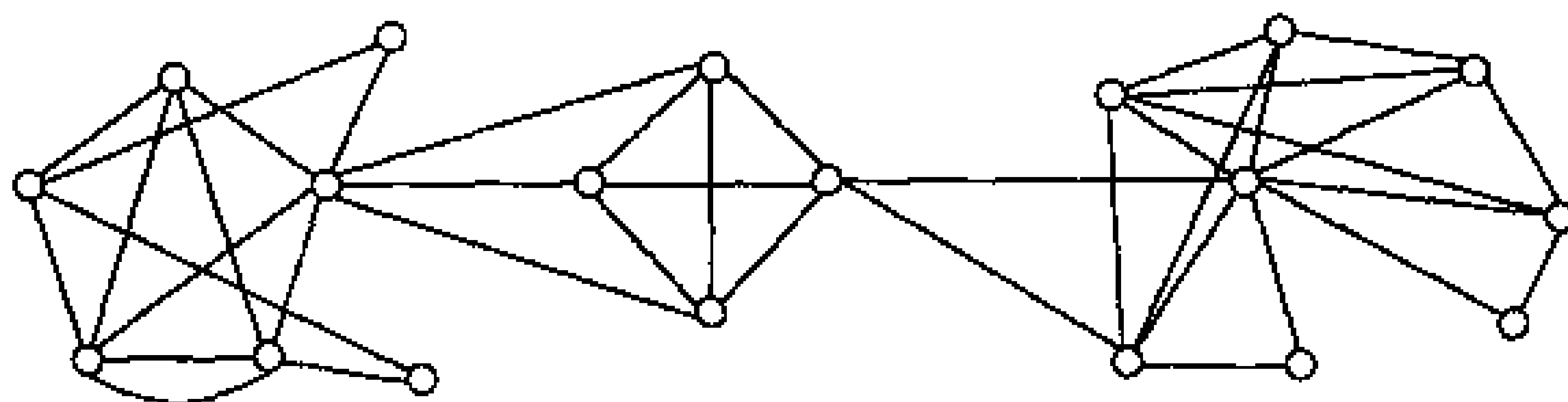
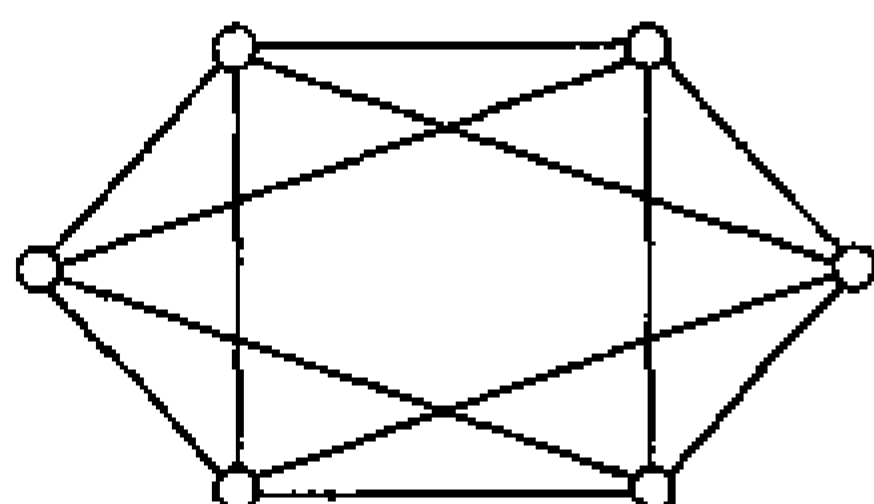
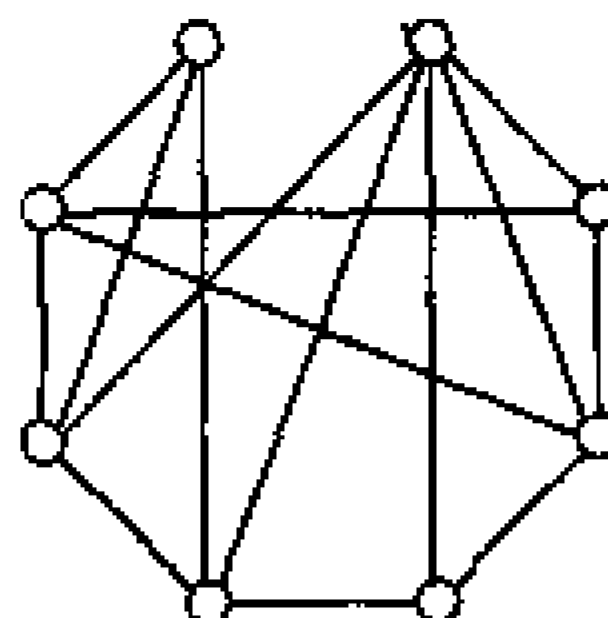


图 4-28 G

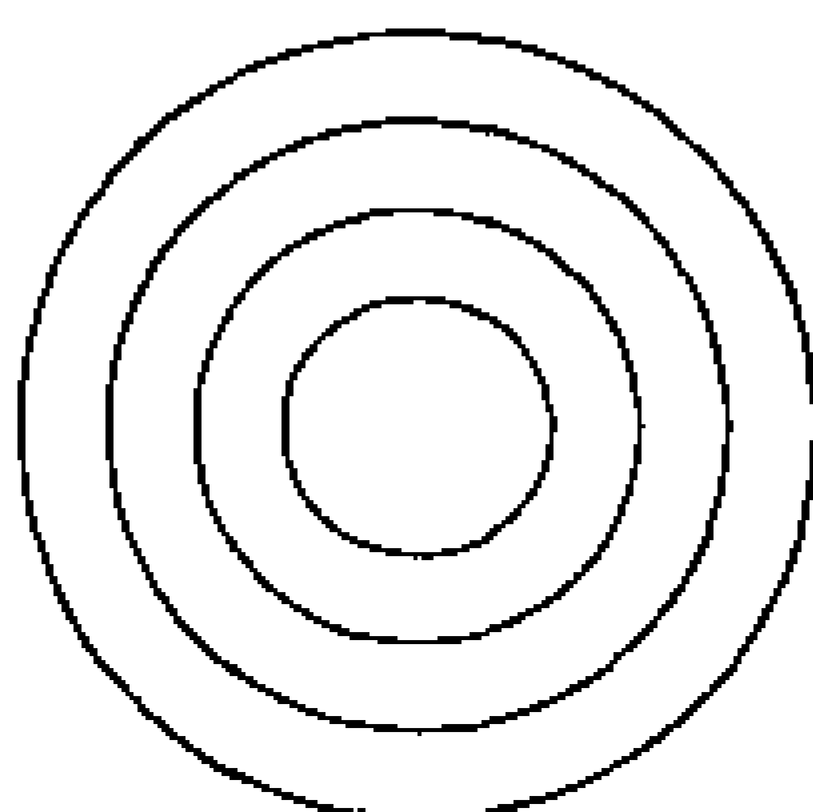


G_1

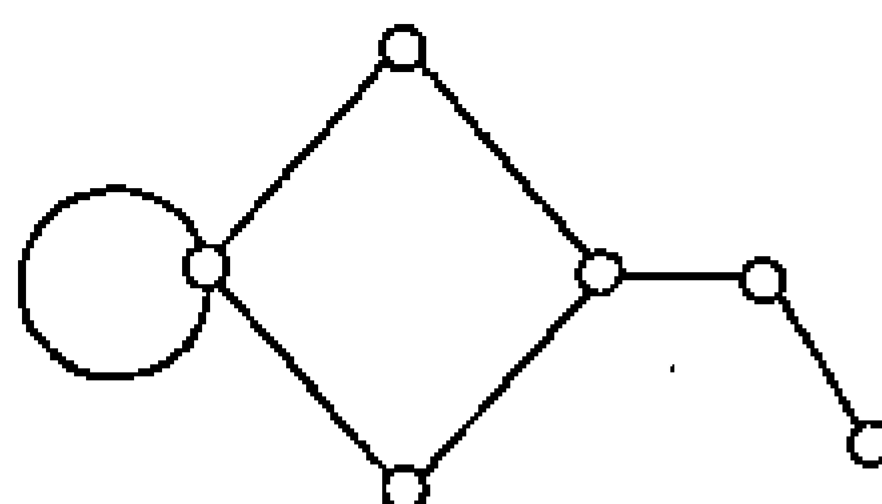


G_2

图 4-29



G_1



G_2

图 4-30

16. 求图 4-31 所示各图的色数.

17. 对 $n \geqslant 3$ 来说,当给圈图 C_n 添加另一个顶点,而且把这个新顶点与 C_n 里 n 个顶点逐个连接时,就得到轮图 W_n. 问 W_n 的色数是多少?

18. 假定当两家电视台相距在 150 英里以内时,它们就不能使用相同的频道,那么对于位于表 4-1 中所示距离的 6 家电视台

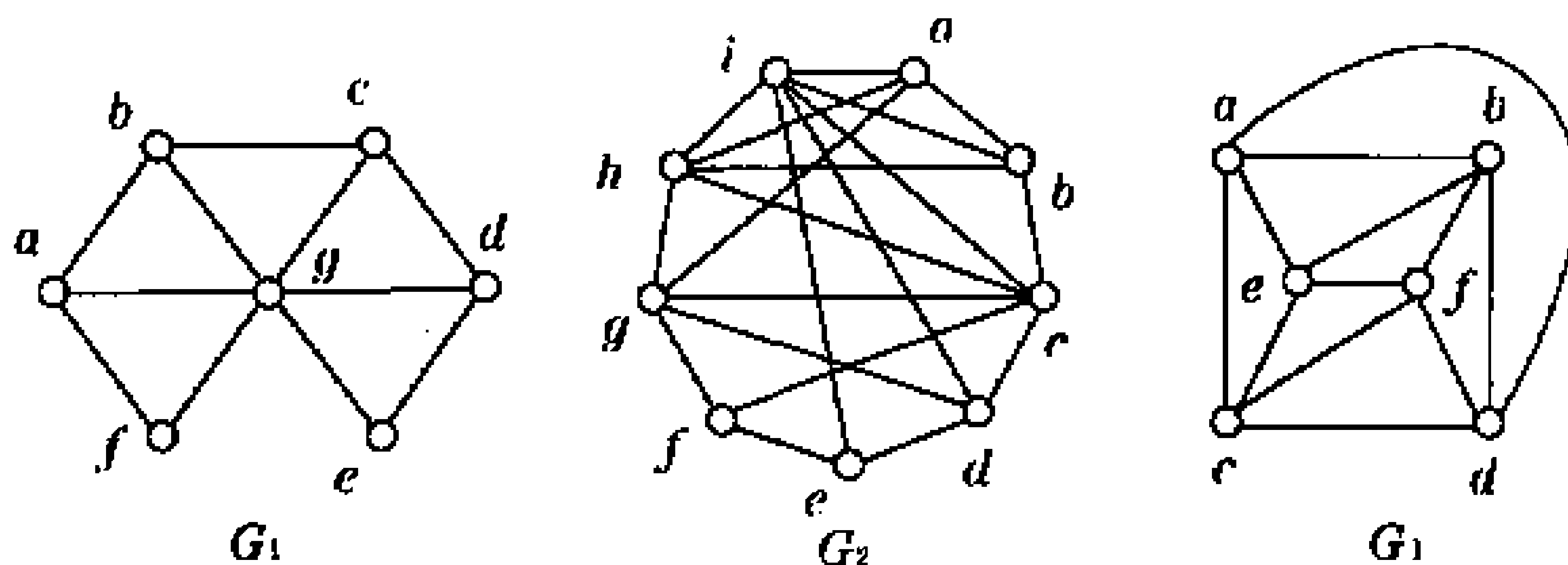


图 4-31

来说,需要多少个不同的频道?

表 4－1

	1	2	3	4	5	6
1	–	85	175	200	50	100
2	85	–	125	175	100	160
3	175	125	–	100	200	250
4	200	175	100	–	210	220
5	50	100	200	210	–	100
6	100	160	250	220	100	–

19. 求下列图的边色数.

(1)K_n　(2)$K_{m,n}$　(3)　C_n　(4)W_n

20. 有 7 个变量出现在计算机程序的循环里.这些变量和必须保存它们的计算步骤是:t:步骤 1 到 6;u:步骤 2;v:步骤 2 到 4;w:步骤 1,3 和 5;x:步骤 1 和 6;y:步骤 3 到 6;z:步骤 4 和 5.在执行期间需要多少个不同的变址寄存器来保存这些变量?

21. 证明:设 G 有 v 个顶点 ϵ 条边,且 G 是自对偶平面图,则 $\epsilon=2v-2$.

22. 证明轮图 W_n 的对偶图 W_n^* 是轮图,且 $W_n\simeq W_n^*$.

23. 证明:一个平面图 G 的对偶图 G^* 是欧拉图当且仅当 G 中每个面均由偶数条边围成.

24. 设 G 是连通的平面图，证明：G 为二分图当且仅当 G 的对偶图为欧拉图.

25. 对于图 4-32 所示的两个图，各自能否找到一条闭折线，使其穿过图中每边一次并且仅一次？为什么？

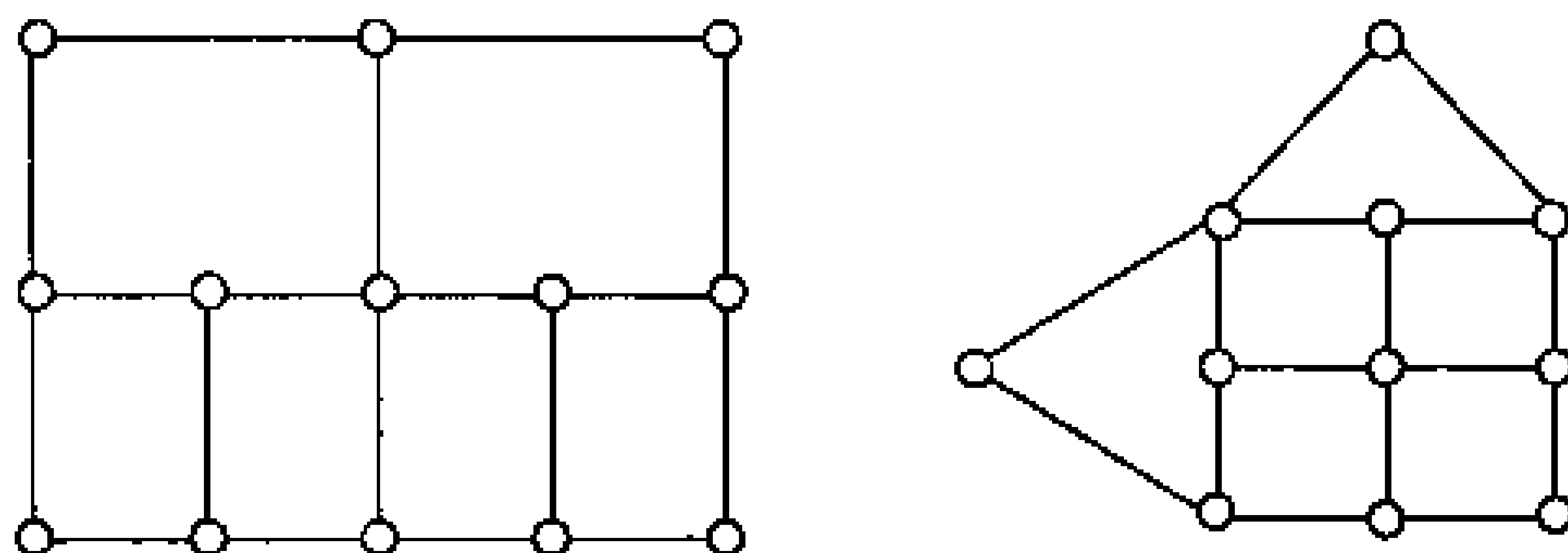


图 4-32

26. 编程解答下列问题

(1)给定学生及其注册课程的表，构造期末考试日程表.

(2)给定各对电视台之间的距离，为这些台分配频率.

(3)给定图 G，求图 G 的色数.

(4)给定图 G，求图 G 的着色.

第五章　独立集与匹配

第一节　独立集

定义 1　设 $G=\langle V,E\rangle$是简单无向图，$S\subseteq V$，$S\neq\emptyset$，若 S 中任何两个顶点都不相邻，则称这个顶点集合 S 为图G 的**独立集**(**independent set**)．若 S 是图G 的独立集，但是任意增加一个顶点就破坏它的独立性，则称这个独立集 S 为**极大独立集**(**maximal independent set**)．独立集 S 称为**最大独立集**(**greatest independent set**)，如果不存在独立集 S'，使$|S'|>|S|$，其中$|S|$为集合 S 的基数．G 的最大独立集 S 的基数$|S|$称为 G 的**独立数**(**independent number**)，记作 $\alpha(G)$．

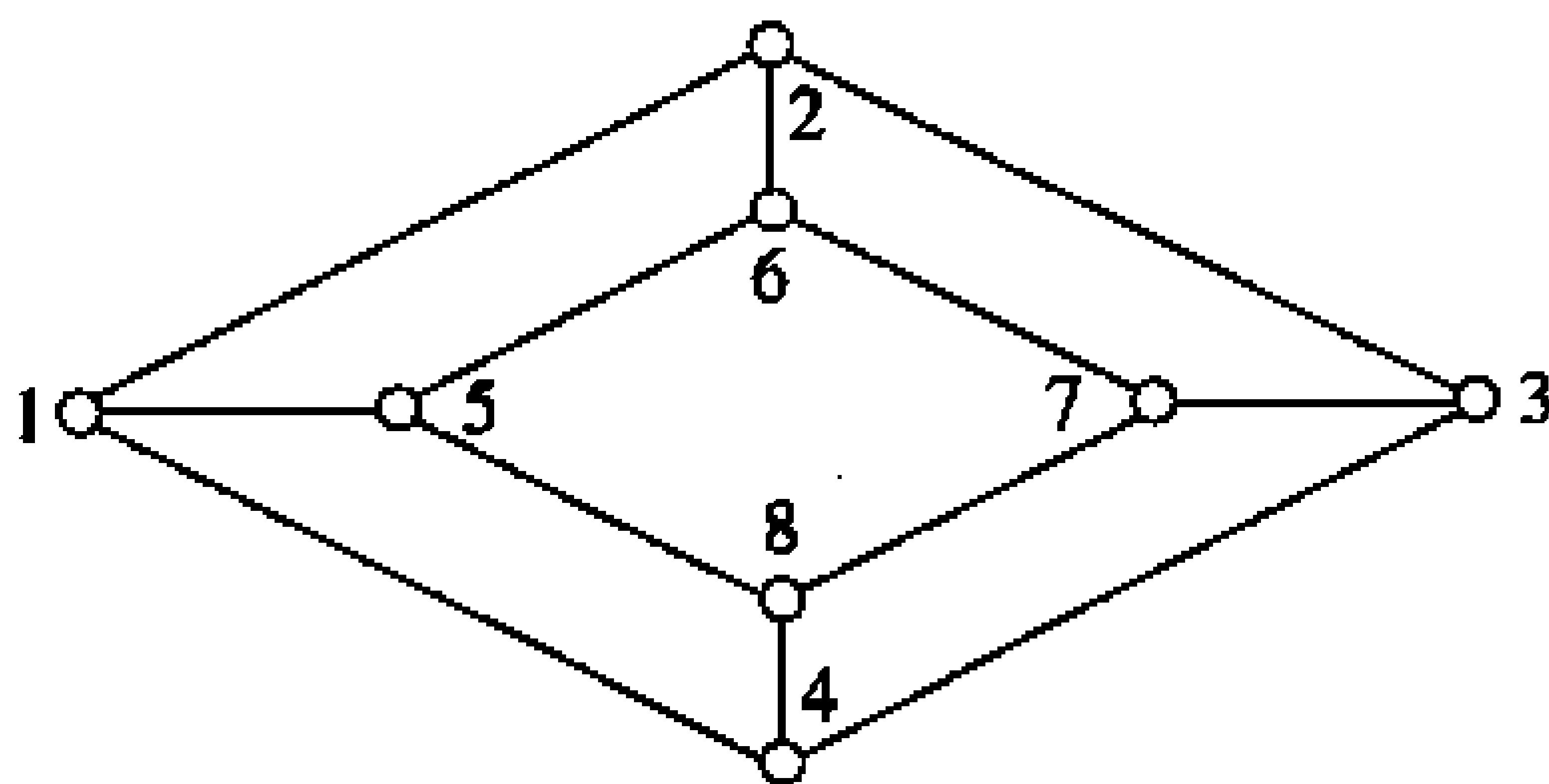


图 5-1　G

简单无向图G 的独立集，实际是对图 G 的顶点进行着色的结果．把图 G 的顶点集 V 划分成若干不相交的子集，每个子集中的各结点着同一色．上述不相交的子集的最少个数即为图 G 的色数．

显然，图 G 的极大独立集不是唯一的，且顶点数目最多的极大独立集是最大独立集．例如，如图 5-1 所示图 G，$\{2,8\}$ 与 $\{2,4,5,7\}$ 都是 G 的极大独立集，同时 $\{2,4,5,7\}$ 也是 G 的最大独立集，因而 G 的独立数 $\alpha(G)=4$．

我们知道，对于简单无向图 $G=\langle V,E\rangle$，G 的邻接矩阵是对称的布尔矩阵．

定义 2 设 $G=\langle V,E\rangle$ 是简单无向图，同时将 G 的邻接矩阵第 i 行与第 j 行，第 i 列与第 j 列互换，称为一次**平移变换**．

显然，平移变换不影响邻接矩阵所表示图 G 的各顶点之间的关系，仅仅改变了 i,j 的编号．也就是说，邻接矩阵的平移变换对应于图中结点的一个重新编号．反之，结点的重新编号对应于邻接矩阵的一系列平移变换．

定理 1 设 $G=\langle V,E\rangle$ 是具有 n 个结点的无向简单图，A 是 G 的邻接矩阵，且 A 具有如下形式：

$$A=\begin{pmatrix} 0 & A_{21}^T \\ A_{21} & A_{22} \end{pmatrix} \tag{1}$$

其中：$A_{22}\in R^{(n-i)\times(n-i)}$

$$A_{21}=\begin{bmatrix} a_{i+1,1} & a_{i+1,2} & \cdots & a_{i+1,i} \\ a_{i+2,1} & a_{i+2,2} & \cdots & a_{i+2,i} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,i} \end{bmatrix} \tag{2}$$

令 $b_j=\sum_{k=1}^{i} a_{jk}(j=i+1,\cdots,n)$，若 $b_j>0(j=i+1,\cdots,n)$ (3)

则其已确定一极大独立集 $S=\{V_1, V_2, \cdots, V_i\}$，其中 $V_t(1\leq t\leq i)$ 为 A 下三角阵的第 t 行．

证明 由矩阵 A 可知，$a_{kj}=0(1\leq j\leq i)$，即可知结点 V_1，$V_2,\cdots,V_i$ 互不相邻接．

考虑 A_{21}，因 $b_j>0(i+1\leq j\leq n)$，则 $a_{j1},a_{j2},\cdots,a_{ji}$ 中必有一元素为 1，不妨设 $a_{jk}=1(1\leq k\leq i)$，即说明 V_j 与 V_k 相邻. 由 $j=\{i+1,i+2,\cdots,n\}$ 的任意性，可知 $\{V_{i+1},V_{i+2},\cdots,V_n\}$ 中所有元素都与 $S=\{V_1,V_2,\cdots,V_i\}$ 相邻接，而 $S=\{V_1,V_2,\cdots,V_i\}$ 中任何两点不邻接. 由极大独立集的定义可知 $S=\{V_1,V_2,\cdots,V_i\}$ 即为图 G 的一个极大独立集. 证毕.

为方便，我们称形如式(1)且满足式(3)的邻接矩阵为**标准型**.

定理 2 设 A 是简单无向图 $G=\langle V,E\rangle$ 的邻接矩阵，则总可以经过若干次平移变换把 A 化为标准型，从而得到图 G 的一个极大独立集.

证明 设一邻接矩阵 A，通过一系列平移变换，可将 A 化为形如式(1)的形式，如果 A 是标准型，即由定理 1 可找出极大独立集 $\{V_1,V_2,\cdots,V_i\}$. 若 A 不是标准型，则 A_{21} 中必有一行元素全为 0，不妨设第 $k(k>i)$ 行元素 $a_{kj}=0(1\leq j\leq i)$. 将第 $i+1$ 项与第 k 项进行平移变换，得到一新的矩阵 A'，再判断 A' 是否为标准型. 我们知道，这种平移变换仅对应于图中结点的一个重新编号，不影响该矩阵所表示图 G 各点之间的邻接关系. 证毕.

下面介绍通过布尔变量的运算找出图 G 的所有极大独立集.

已知简单无向图 $G=\langle V,E\rangle$，且 $V=\{V_1,V_2,\cdots,V_n\}$，我们作如下约定：

(1) G 的每个顶点 V_i 当作一个布尔变量；

(2) $V_i\wedge V_j$ 表示包含 V_i 和 V_j；

(3) $V_i\vee V_j$ 表示以下三种情形：或者包含一顶点 V_i；或者包含一顶点 V_j；或者包含 V_i,V_j 两点.

根据以上约定，我们列举布尔积 $\wedge$ 和布尔和 $\vee$ 的几个性质：

(1) $V_i\wedge V_j=V_j\wedge V_i$，$V_i\vee V_j=V_j\vee V_i$（交换）

(2) $(V_i\wedge V_j)\wedge V_t=V_i\wedge(V_j\wedge V_t)$（结合）

$(V_i \vee V_j) \vee V_t = V_i \vee (V_j \vee V_t)$

(3) $V_i \wedge V_i = V_i$，$V_i \vee V_i = V_i$（等幂）

(4) $V_i \wedge (V_j \vee V_t) = (V_i \wedge V_j) \vee (V_i \wedge V_t)$（分配）

$V_i \vee (V_j \wedge V_t) = (V_i \vee V_j) \wedge (V_i \vee V_t)$

(5) $V_i \vee (V_i \wedge V_j) = V_i$，$V_i \wedge (V_i \vee V_j) = V_i$（吸收）

(6) $(\overline{V_i \wedge V_j}) = \overline{V}_i \vee \overline{V}_j$，$(\overline{V_i \vee V_j}) = \overline{V}_i \wedge \overline{V}_j$（德·摩根律）

这些运算性质，都可以通过列真值表加以证明，比如，表 5－1 的真值表证明了性质(5).

表 5－1

V_i	V_j	$V_i \wedge V_j$	$V_i \vee V_j$	$V_i \vee (V_i \wedge V_j)$	$V_i \wedge (V_i \vee V_j)$
T	T	T	T	T	T
T	F	F	T	T	T
F	T	F	T	F	F
F	F	F	F	F	F

显然，图 G 的过顶点 V_i，V_j 的边 $\{V_i, V_j\}$ 对应一布尔积 $V_i \wedge V_j$. 作布尔表达式 $\varphi = \bigvee_{\{V_i, V_j\} \in E} (V_i \wedge V_j)$，即 φ 中的每一项 $V_i \wedge V_j$ 对应于 G 的一条边，$\vee$ 是对所有的边求布尔和. 由德·摩根律，我们有 $\bar{\varphi} = \bigwedge_{\{V_i, V_j\} \in E} (\overline{V}_i \vee \overline{V}_j)$.

设 $\bar{\varphi} = \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k$，$\varphi$ 和 $\bar{\varphi}$ 都是含有布尔变量 $V_1, V_2, \cdots V_n$ 的表达式. 因 G 的极大独立集不包含任何一边的两个端点，故表达式 φ 在任一极大独立集上取布尔值 0(F)；反之，使 φ 取值 0 的点集是独立集. 即布尔表达式 φ 取布尔值 0 是独立集的充要条件. 换句话说，使 $\bar{\varphi}$ 取布尔值 1(T)是独立集的充要条件. 由于 $\bar{\varphi} = \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k$，只要其中任一项为 1，则 $\bar{\varphi} = 1$. 故分别使 $\varphi_1, \varphi_2, \cdots, \varphi_k$ 取布尔值 1 的点集都是极大独立集.

例 1 通过布尔变量的运算，求图 5-2 所示图 G 的极大独立

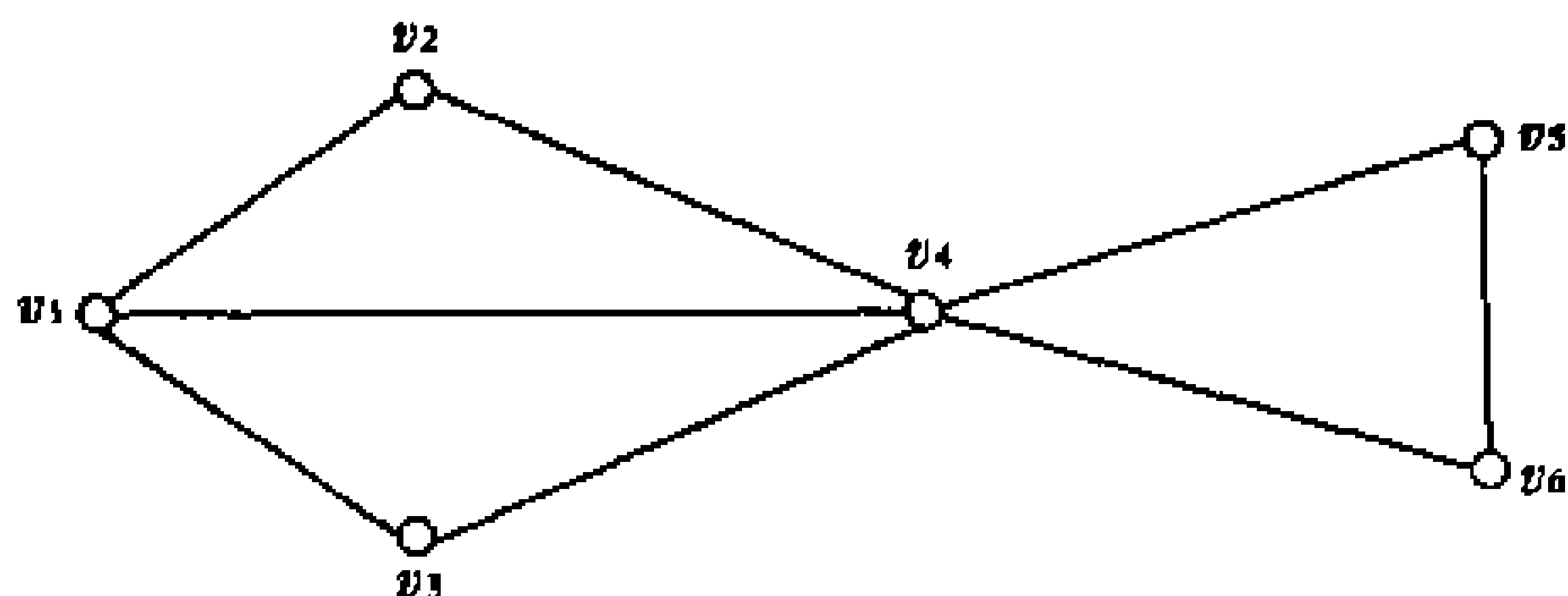


图 5-2　G

集.

解　$\varphi=(v_1\wedge v_2)\vee(v_1\wedge v_3)\vee(v_1\wedge v_4)\vee(v_2\wedge v_4)\vee(v_3\wedge v_4)\vee(v_4\wedge v_5)\vee(v_4\wedge v_6)\vee(v_5\wedge v_6)$

$$\bar\varphi=(\bar v_1\vee\bar v_2)\wedge(\bar v_1\vee\bar v_3)\wedge(\bar v_1\vee\bar v_4)\wedge(\bar v_2\vee\bar v_4)\wedge(\bar v_3\vee\bar v_4)\wedge(\bar v_4\vee\bar v_5)\wedge(\bar v_4\vee\bar v_6)\wedge(\bar v_5\vee\bar v_6)$$

$$\begin{aligned}(\bar v_1\vee\bar v_2)\wedge(\bar v_1\vee\bar v_3)&=[((\bar v_1\vee\bar v_2)\wedge\bar v_1]\vee[(\bar v_1\vee\bar v_2)\wedge\bar v_3]\\&=[(\bar v_1\wedge\bar v_1)\vee(\bar v_2\wedge\bar v_1)]\vee[(\bar v_1\wedge\bar v_3)\vee(\bar v_2\wedge\bar v_3)]\\&=\bar v_1\vee(\bar v_2\wedge\bar v_1)\vee(\bar v_1\wedge\bar v_3)\vee(\bar v_2\wedge\bar v_3)\\&=\bar v_1\vee(\bar v_2\wedge\bar v_3)\end{aligned}$$

$$(\bar v_1\vee\bar v_4)\wedge(\bar v_2\vee\bar v_4)=\bar v_4\vee(\bar v_1\wedge\bar v_2)$$

$$(\bar v_3\vee\bar v_4)\wedge(\bar v_4\vee\bar v_5)=\bar v_4\vee(\bar v_3\wedge\bar v_5)$$

$$(\bar v_4\vee\bar v_6)\wedge(\bar v_5\vee\bar v_6)=\bar v_6\vee(\bar v_4\wedge\bar v_5)$$

所以

$$\begin{aligned}\bar\varphi&=[\bar v_1\vee(\bar v_2\wedge\bar v_3)]\wedge[\bar v_4\vee(\bar v_1\wedge\bar v_2)]\wedge[\bar v_4\vee(\bar v_3\wedge\bar v_5)]\wedge[\bar v_6\vee(\bar v_4\wedge\bar v_5)]\\&=(\bar v_1\wedge\bar v_4\wedge\bar v_6)\vee(\bar v_1\wedge\bar v_4\wedge\bar v_5)\vee(\bar v_2\wedge\bar v_3\wedge\bar v_4\wedge\bar v_6)\vee(\bar v_2\wedge\bar v_3\wedge\bar v_4\wedge\bar v_5)\vee\\&\quad(\bar v_1\wedge\bar v_2\wedge\bar v_3\wedge\bar v_5\wedge\bar v_6)\end{aligned}$$

故得下面的各极大独立集:

$\{v_2,v_3,v_5\},\{v_2,v_3,v_6\},\{v_1,v_5\},\{v_1,v_6\},\{v_4\}$.

定义 2 设 $G=\langle V,E\rangle$是无向简单图，$S\subseteq V$，$S\neq\varnothing$. 若 E 中每条边都与 S 中某点关联，则称 S 为 G 的**点覆盖(vertex covering)**. 如果 G 中的任何异于 S 的点覆盖 S'，均有 $|S'|\geqslant|S|$，则称 S 为 G 的**最小点覆盖(smallest vertex covering)**. 最小点覆盖 S 的基数 $|S|$ 称为 G 的**点覆盖数(vertex covering number)**，记作 $\beta(G)$. 点覆盖 S 称为**极小点覆盖(minimal vertex covering)**，若对任何 $x\in S$，$S-\{x\}$都不是点覆盖.

定理 3 设 $G=\langle V,E\rangle$是简单无向图，$S\subseteq V$，$S\neq\varnothing$. 则 S 是 G 的独立集$\Leftrightarrow V-S$ 是 G 的点覆盖.

证明 S 是 G 的独立集$\Leftrightarrow G$ 中每条边的两端点都不同时属于 $S\Leftrightarrow G$ 中每条边至少有一端点在 $V-S$ 中$\Leftrightarrow V-S$ 是 G 的点覆盖.

推论 1 S 是 G 的极大独立集$\Leftrightarrow V(G)-S$ 是 G 的极小点覆盖.

推论 2 $\alpha(G)+\beta(G)=|V(G)|$.

证明 设 S_1 是 G 的最大独立集，S_2 是 G 的最小点覆盖，由定理 3 知 $V(G)-S_1$ 是点覆盖，$V(G)-S_2$ 是独立集. 因而

$$|V(G)|-\alpha(G)=|V(G)-S_1|\geqslant\beta(G)$$

$$|V(G)|-\beta(G)=|V(G)-S_2|\leqslant\alpha(G)$$

由此得
$$\alpha(G)+\beta(G)=|V(G)|.$$

定义 3 设 $G=\langle V,E\rangle$是无向简单图，$L\subseteq E$，$L\neq\varnothing$. 若 G 的每个顶点都与 L 中某条边关联，则称 L 为 G 的**边覆盖(edge covering)**. 如果 G 的任何异于 L 的边覆盖 L'，均有 $|L'|\geqslant|L|$，则称 L 为 G 的**最小边覆盖(smallest edge covering)**. G 的最小边覆盖 L 的基数 $|L|$ 称为 G 的**边覆盖数(edg covering number)**，记作 $\beta'(G)$.

第二节 独立集的应用

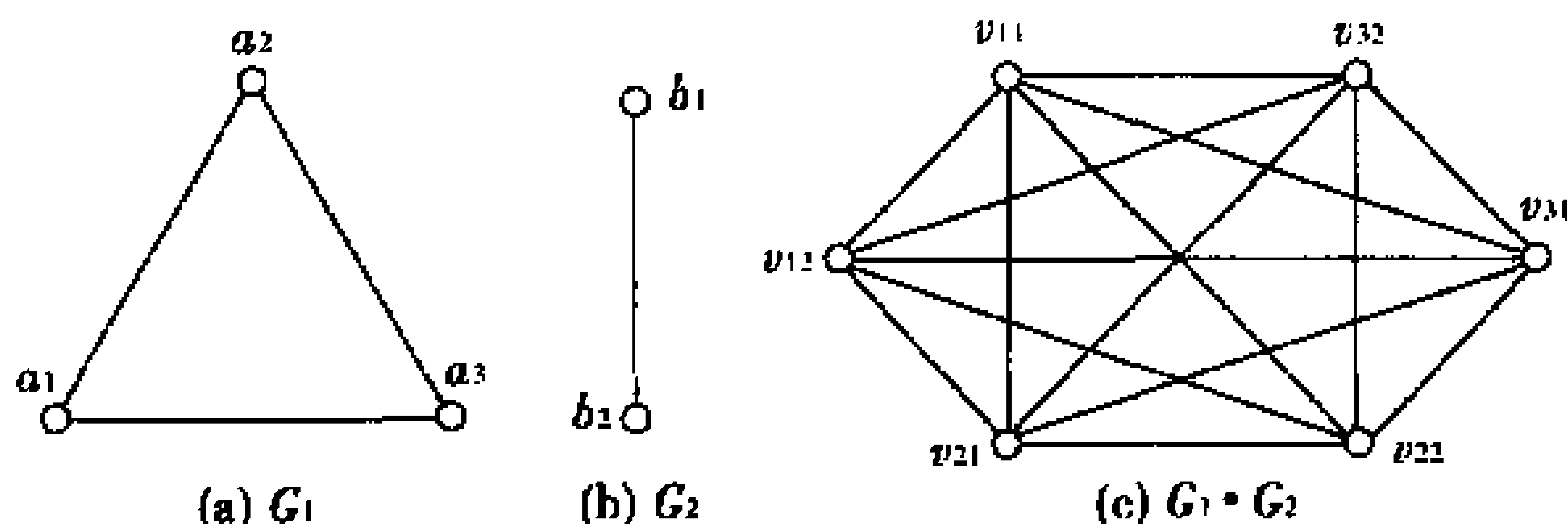
独立集有着广泛的应用，尤其广泛用于系统故障诊断领域. 下面介绍两个具体的应用实例.

定义 1 设 $G_1=\langle V_1,E_1\rangle$ 和 $G_2=\langle V_2,E_2\rangle$ 是两个无向简单图，其中 $V_1=\{a_1,a_2,\cdots,a_n\}$，$V_2=\{b_1,b_2,\cdots,b_m\}$. 图 $G=G_1\cdot G_2=\langle V,E\rangle$ 称为图 G_1 与 G_2 的乘积，若满足：

(1) $V=\{\langle a_i,b_j\rangle \mid a_i\in V_1,b_j\in V_2\}$；

(2) $adj(\langle a_i,b_j\rangle)=\{\langle a_k,b_j\rangle \mid a_k\in adj(a_i)\}\cup\{\langle a_i,b_t\rangle \mid b_t\in adj(b_j)\}\cup\{\langle a_k,b_t\rangle \mid a_k\in adj(a_i),b_t\in adj(b_j)\}$ 其中，$adj(v_i)$ 表示与 v_i 点相邻的点的集合.

当然，$\langle a_i,b_j\rangle$ 不是边，而是图 $G_1\cdot G_2$ 的一个顶点. 例如，图 5-3(a)为 G_1，5-3(b)为 G_2，5-3(c)为 $G_1\cdot G_2$.

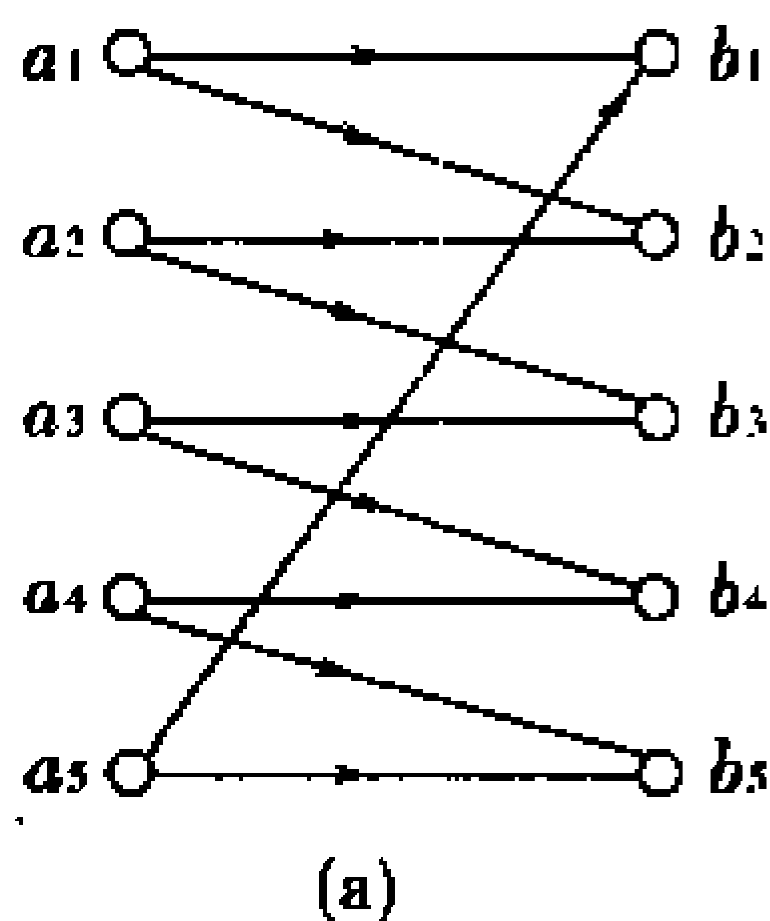


(a) G_1　(b) G_2　(c) $G_1\cdot G_2$

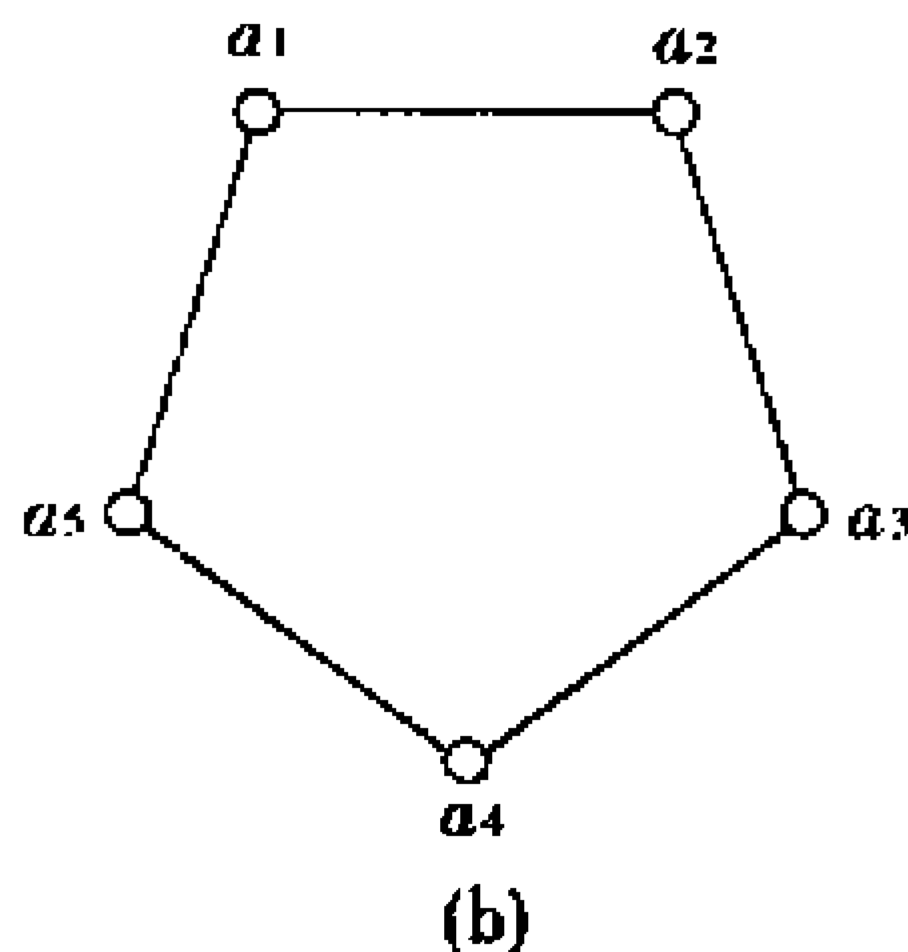
图 5-3

下面我们来看一个独立集应用的例子.

图 5-4 表示某一通讯道里，输入端有五个符号允许输入，设为 a_1,a_2,a_3,a_4,a_5，输出端可以获得 b_1,b_2,b_3,b_4,b_5 五种讯息. 由于有干扰，若输入端为 a_1 时，输出端可能为 b_1，也可能为 b_2；同样，输入端为 a_2 时，输出端可能为 b_2 或 b_3；等等. 这样，从输出端



(a)



(b)

图 5-4

输出的讯息不能正确地断定输入端的讯息. 为了正确无误地从输出端得到输入端的讯息,输入端只能选取 a_1,a_2,a_3,a_4,a_5 中的若干符号,而不能是它们的全体.

图 5-4(b)说明 a_1 可能和 a_2 发生错乱,也可能与 a_5 发生错乱,a_2 可能和 a_1 也可能和 a_3 发生错乱,等等. 所以为了使输出端输出的讯息正确无误地反映输入端的讯息,问题导致求图 5-4(b)的极大独立集. 例如选取$\{a_1,a_3\}$,$\{a_1,a_4\}$,$\{a_2,a_5\}$等.

如果要传输两个以上的信息时,就得采用由几个符号组成的码.

从表 5-2 可以看出采用 $a_1a_1,a_2a_3,a_3a_5,a_4a_2,a_5a_4$ 这五个字符串可明确无误地表达 5 种讯息,即输出端讯息没有混淆不清的. 可能有 20 个输出各不相同.

表 5-2

输入端讯息	输出端讯息			
a_1a_1	b_1b_1	b_1b_2	b_2b_1	b_2b_2
a_2a_3	b_2b_3	b_2b_4	b_3b_3	b_3b_4
a_3a_5	b_3b_5	b_3b_1	b_4b_5	b_4b_1
a_4a_2	b_4b_2	b_4b_3	b_5b_2	b_5b_3
a_5a_4	b_5b_4	b_5b_5	b_1b_4	b_1b_5

为了找出两个字符组成的使输出端给出明确无误的讯息问题，导致求图 $G\cdot G$ 的极大独立集.类似的理由，如果必要时可以求 $G\cdot G\cdot G$ 的极大独立集，等等.

下面再来讨论一个独立集的应用实例.

某大型商场为加强经营管理，对商品的零售收入实行统一收款制度.为了使顾客在任何一个货架前都能看到收款台，问收款台应设置在什么地方并且至少要设置多少个收款台？这个问题称为**收款台的设置问题**.

有许多实际问题都可以归结为这个问题.例如，某军事基地固定岗哨的设置问题，某道路交通系统交通监控器的设置问题等.

构作简单无向图 $G=\langle V,E\rangle$，该商场两排货架之间的通道为 G 的边，通道交叉处为 G 的顶点.为使顾客在任何一个货架前都能看到收款台，从尽可能减少设置收款台的数目来说，收款台应设置在通道的交叉处.于是收款台的设置问题就归结为在 G 中找出一个最小点覆盖来.

我们知道，如果找到 G 的最大独立集 S，则 $G-S$ 是最小点覆盖.因而，收款台的设置问题，实际上就是在 G 中找出一个最大独立集.

例 1 求图 5-5 所示图 G 的最小点覆盖.

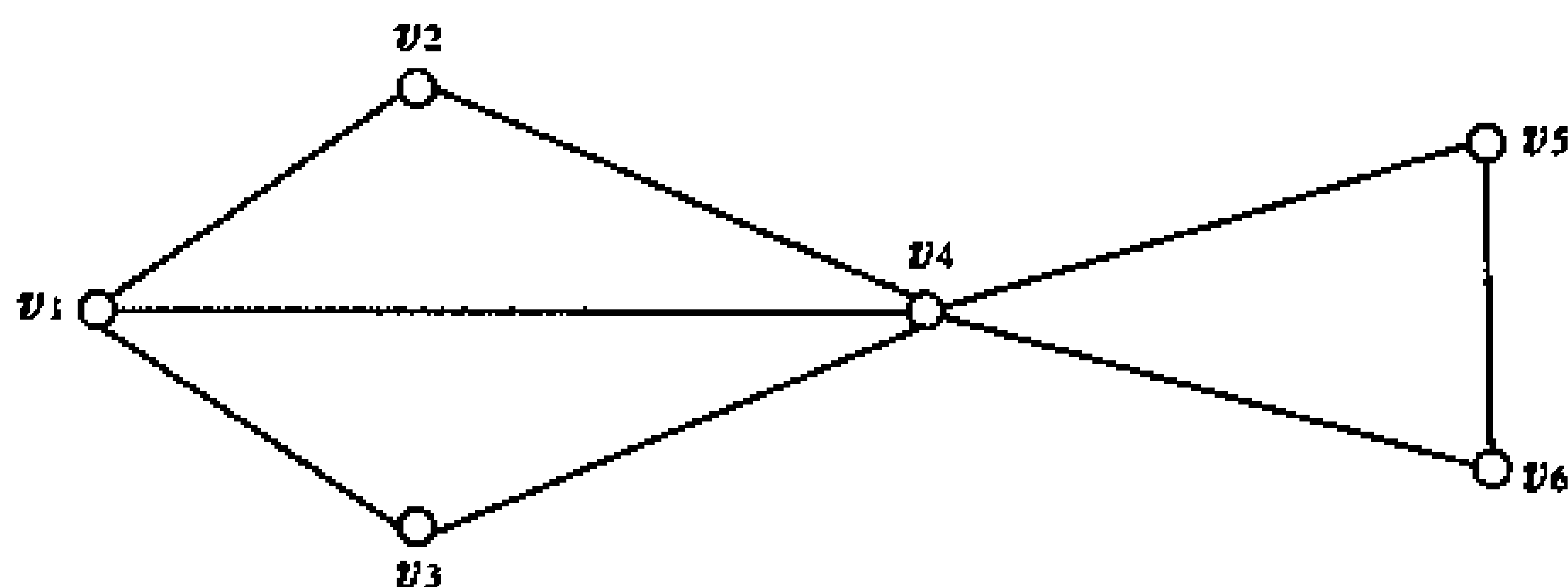


图 5-5 G

解 G 的邻接矩阵 A 为

$$A = \begin{array}{c|cccccc|} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline v_1 & 0 & 1 & 1 & 1 & 0 & 0 \\ v_2 & 1 & 0 & 0 & 1 & 0 & 0 \\ v_3 & 1 & 0 & 0 & 1 & 0 & 0 \\ v_4 & 1 & 1 & 1 & 1 & 1 & 1 \\ v_5 & 0 & 0 & 0 & 1 & 0 & 1 \\ v_6 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

把 A 的 1 与 6 项作平移变换得 A_1;把 A 的 1 与 5 项作平移变换得 A_2.

$$A_1 = \begin{array}{c|ccc:ccc|} & v_6 & v_2 & v_3 & v_4 & v_5 & v_1 \\ \hline v_6 & 0 & 0 & 0 & 1 & 1 & 0 \\ v_2 & 0 & 0 & 0 & 1 & 0 & 1 \\ v_3 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hdashline v_4 & 1 & 1 & 1 & 1 & 1 & 1 \\ v_5 & 1 & 0 & 0 & 1 & 0 & 0 \\ v_1 & 0 & 1 & 1 & 1 & 0 & 0 \end{array}$$

$$A_2 = \begin{array}{c|ccc:ccc|} & v_5 & v_2 & v_3 & v_4 & v_1 & v_6 \\ \hline v_5 & 0 & 0 & 0 & 1 & 0 & 1 \\ v_2 & 0 & 0 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hdashline v_4 & 1 & 1 & 1 & 1 & 1 & 1 \\ v_1 & 0 & 1 & 1 & 1 & 0 & 0 \\ v_6 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

A_1 和 A_2 都是标准形式，从而可知 $\{v_2, v_3, v_6\}$ 与 $\{v_2, v_3, v_5\}$ 都是 G 的最大独立集. 因此，$\{v_1, v_4, v_5\}$ 与 $\{v_1, v_4, v_6\}$ 都是 G 的最小点覆盖.

设 $G=\langle V,E\rangle$ 是简单无向图，$T\subseteq V$，$T\neq\varnothing$，若 T 中任意两个顶点都相邻，则称 T 是图 G 的**团**(clique). 若 T 是图 G 的团，但是任意增加一个新顶点后，它就不成为团，则称 T 是图 G 的**极大团**.

一个图的团的概念在下述意义下与独立集是"互补的". 设 $G=\langle V,E\rangle$ 是简单无向图，基中 $V=\{1,2,\cdots,n\}$. $\overline{G}=\langle V,\overline{E}\rangle$ 是 G 的补图，其中顶点 i 与 j 在 $\overline{G}$ 中相邻，当且仅当顶点 i 与 j 在 G 中不相邻. 从独立集与团的概念知，S 是 G 的独立集当且仅当 S 是 $\overline{G}$ 的团；同时，S 是 G 的极大独立集当且仅当 S 是 $\overline{G}$ 的极大团. 因而，为求 G 的极大独立集，转化为求 $\overline{G}$ 的极大团. 当然，一般情况下，求图的团也是非常困难的.

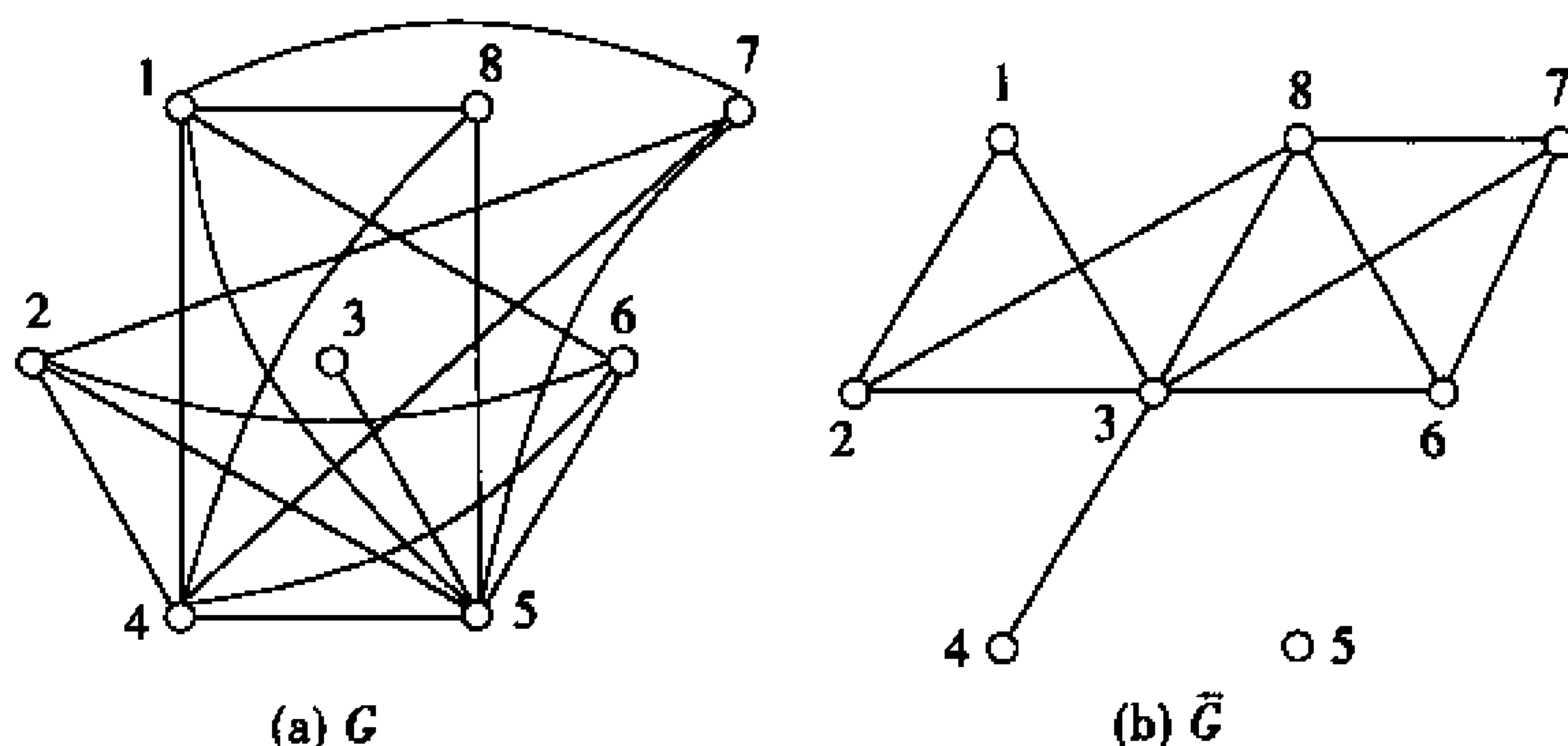


图 5-6

例 2 求图 5-6(a)所示图 G 的极大独立集.

解 图 G 的补图 $\overline{G}$ 如图 5-6(b)所示. $\overline{G}$ 中全部极大完全图的顶点集合依次为 $\{5\}$, $\{3,4\}$, $\{2,3,8\}$, $\{1,2,3\}$, $\{3,6,7,8\}$. 因此，这些集合也恰是 G 的全部极大独立集.

第三节　支配集

定义 1　设图 $G=\langle V,E\rangle$ 是简单无向图，$S\subseteq V$，$S\neq\varnothing$，若对于 $\forall x\in V-S$，x 都与 S 里至少一个顶点相邻，则称 S 是图 G 的**支配集**（**dominating set**）. S 是图 G 的支配集，若 S 的任何真子集都不是支配集，则称 S 为图 G 的**极小支配集**（**minimal dominating set**）. S 是图 G 的支配集，若不存在任何其它支配集 S'，使得 $|S'|<|S|$，则称 S 是图 G 的**最小支配集**（**smallest dominating set**）. 若 S 是图 G 的最小支配集，则称 $|S|$ 为图 G 的**支配数**（**dominating number**），记作 $\gamma(G)$.

例如，图 5-7 所示中，$S_1=\{3,6,9,10\}$ 和 $S_2=\{2,7,11\}$ 是图 G 的两个支配集，用黑实点表示. 因为 S_2 是最小支配集，所以 $\gamma(G)=3$.

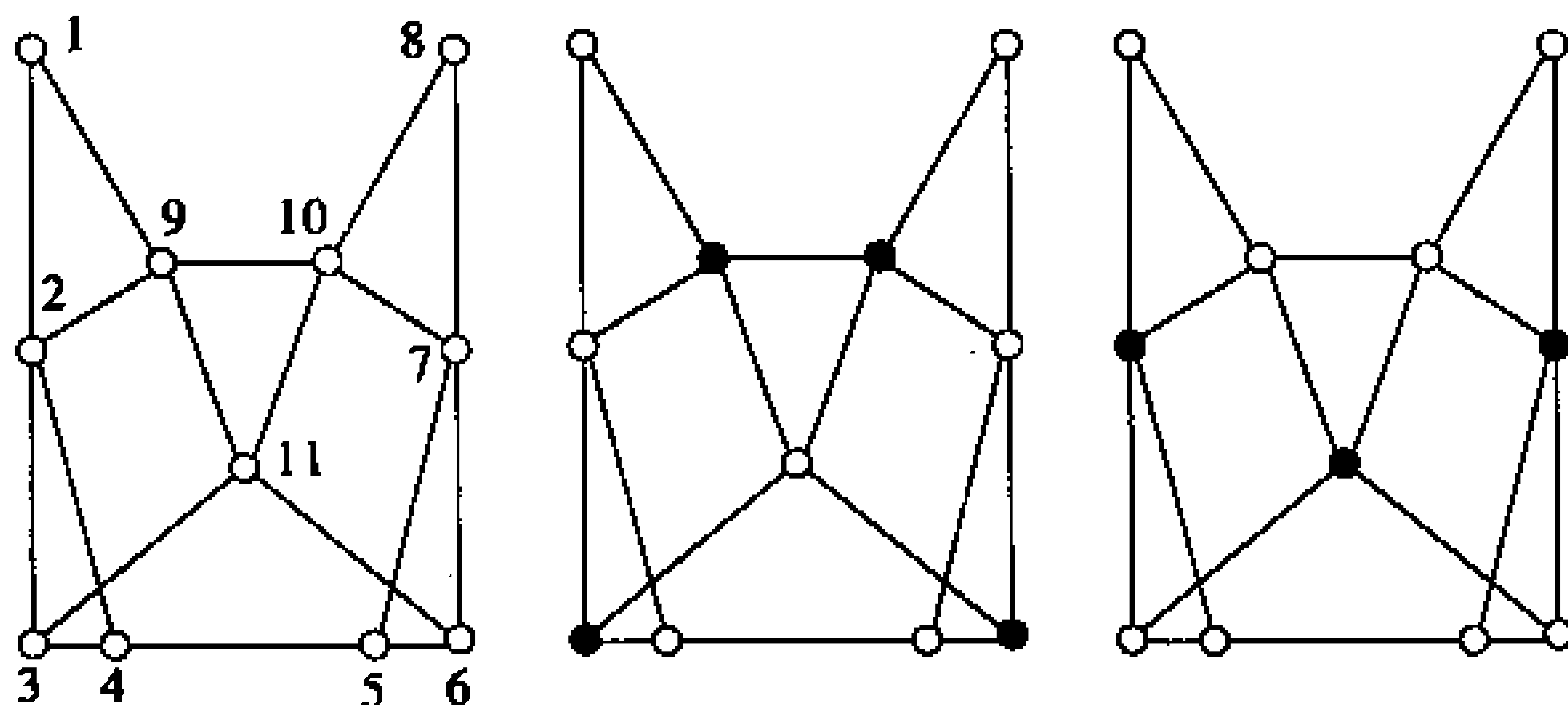
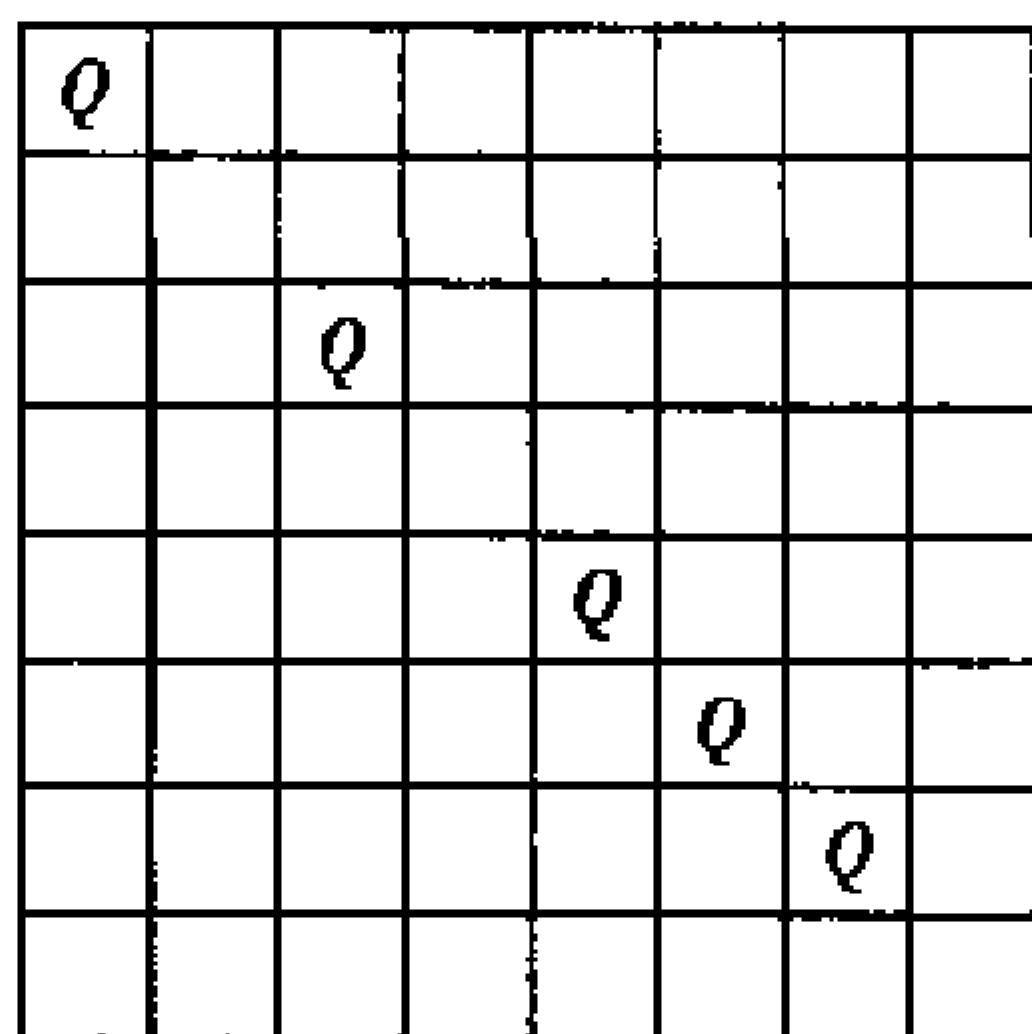


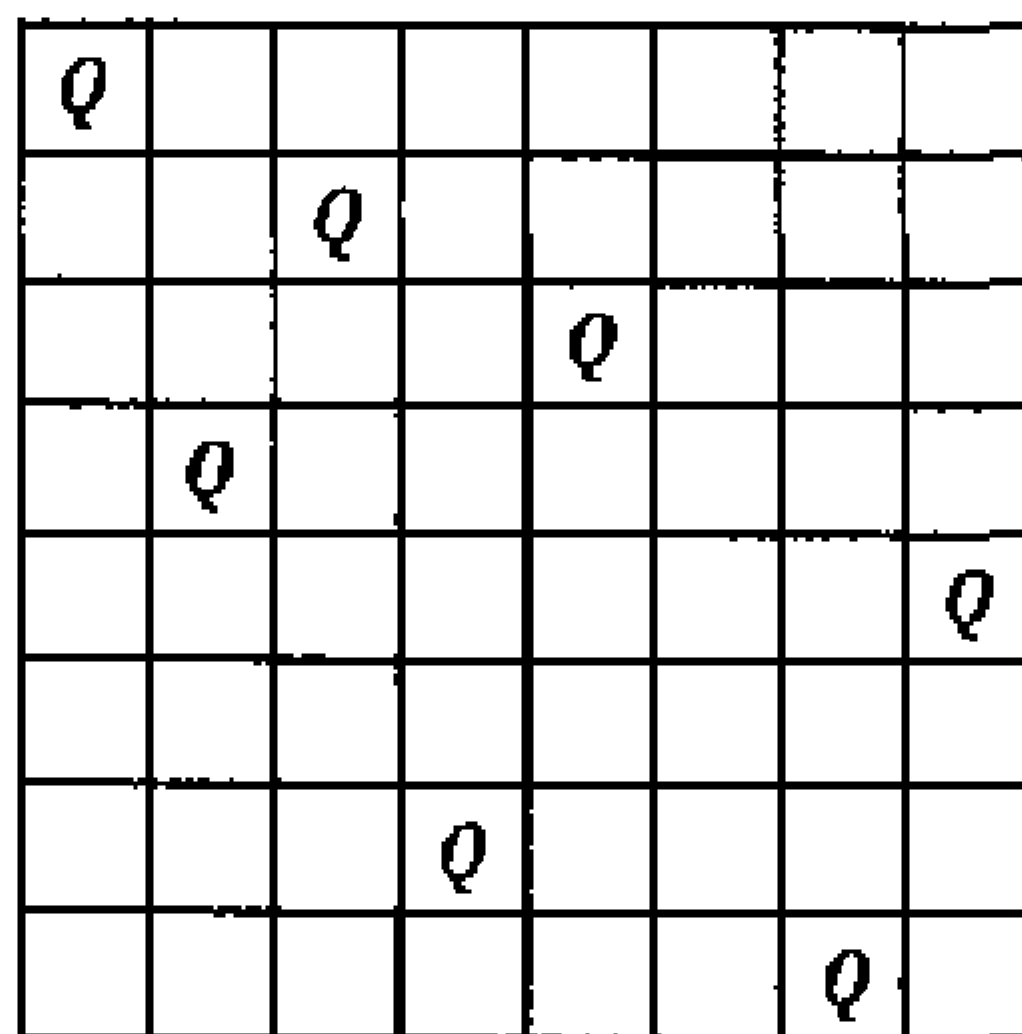
图 5-7

在国际象棋的比赛中，首先出现了支配集的概念. 一个 8×8 的棋盘具有处在 8×8 配置下的 64 个格子. 在所给某个位置的皇后控制着同行、同列以及包含这个格子的两条斜线上的所有格子. 1862 年，De Jaenisch 考虑了控制整个棋盘所需要的最少的皇后个

数问题，这种皇后的最少个数为 5. 图 5-8(a)显示了一种放置方法. 若要求任两个皇后都不相互攻击，即任两个皇后都不在同一行、同一列或一斜线上，那么这种皇后的最少个数为 7，图 5-8(b)显示了一种放置方法.



(a)



(b)

图 5-8

上述棋盘问题和图的支配集之间有着直接的联系. 作简单图 G，图 G 的顶点集与棋盘上的 64 个格子一一对应，且两个顶点在 G 中相邻当且仅当两个对应格子中的一个格子可以由位于另一个格子中的皇后控制. 则支配棋盘中全部格子的皇后的最少个数为 $\gamma(G)$. 支配棋盘中全部方格的互不攻击的皇后的最少个数就是 G 的独立支配集的最小基数.

定理 1 图 G 的支配集 S 是 G 的极小支配集当且仅当 S 中每个顶点 x 满足下列性质之一：

(1) 存在 $y \in V(G)-S$ 使得 $N(y)\cap S=\{x\}$，其中 $N(y)$ 为 y 的邻接点集合.

(2) $S\cap N(x)=\varnothing$，即 S 中所有顶点不与 x 邻接.

证明 如果 S 中每个顶点至少满足(1)和(2)中一个，则 $S-\{x\}$ 就不是 G 的支配集，从而，S 是 G 的一个极小支配集.

反之,假设 S 是 G 的极小支配集.那么,对每个 $x\in S$,集 $S-\{x\}$就不是 G 的支配集.所以存在顶点 $y\in V(G)-(S-\{x\})$,使得没有 $S-\{x\}$中顶点与 y 邻接.如果 $y=x$,则 S 中没有顶点和 x 邻接.假设 $y\neq x$,因为 S 是 G 的支配集且 $y\notin S$,所以顶点 y 至少与 S 中一个顶点邻接.然而,y 不与 $S-\{x\}$中顶点邻接,从而 $N(y)\cap S=\{x\}$.

定理 2 如果 G 是没有孤立结点的图,且 S 是 G 的极小支配集,那么 $V(G)-S$ 也是 G 的支配集.

证明 设 $x\in S$,则 x 至少具备定理 1 中的两个性质之一.首先,假设存在顶点 $y\in V(G)-S$ 使得 $N(y)\cap S=\{x\}$,则 x 邻接到 $V(G)-S$ 中某个顶点.其次,假设 S 中没有顶点和 x 邻接,那么 x 是 S 导出子图中的孤立顶点,因为 x 在 G 中不是孤立顶点,所以顶点 x 邻接到 $V(G)-S$ 中某个顶点.这样 $V(G)-S$ 是 G 的支配集.

推论 1 如果 G 是 n 阶无孤立结点的图,则 $\gamma(G)\leqslant\frac{n}{2}$.

证明 设 S 是 G 的极小支配集,由定理 2 知,$V(G)-S$ 也是 G 的支配集,这样有

$$\gamma(G)\leq\min\{|S|,|V(G)-S|\}\leq\frac{n}{2}.$$

定理 3 如果图 G 不含孤立结点,则存在 G 的最小支配集 S 使得对 S 中每个顶点 x,存在 $V(G)-S$ 中的顶点 y 满足:$N(y)\cap S=\{x\}$,其中 $N(y)$为结点 y 的邻接点集合.

证明 用反证法.用 $G[S]$表示 S 的导出子图.在 G 的全部最小支配集中,设 S 是使 $G[S]$满足$|E(G[S])|$达到最大的一个最小支配集.假设定理结论不成立,S 至少包含一个不具备上述性质的顶点 x,即对 $\forall y\in V(G)-S,N(y)\cap S\neq\{x\}$.由定理 1,$x$ 是 $G[S]$的孤立结点.又因为 S 为 G 的最小支配集,所以对 $\forall y\in$

$V(G)-S, N(y)\cap S\neq\emptyset$，即与 x 邻接的 $V(G)-S$ 中的每个顶点一定与 S 中的另外一个顶点邻接．由于 G 不含孤立顶点，所以 x 与 $V(G)-S$ 中的某个顶点 y 邻接，故 $(S-\{x\})\cup\{y\}$ 是 G 的最小支配集，且它的导出子图中至少包含一条与 y 关联的边，要比 $G[S]$ 中的边数多，与 S 的选取方式矛盾．

定理 4　如果 G 是 n 阶图，则

$$\left\lceil\frac{n}{1+\Delta(G)}\right\rceil\leq\gamma(G)\leq n-\Delta(G).$$

证明　设 S 是 G 的最小支配集，则

$$V(G)-S\subseteq\bigcup_{x\in S}N(x).$$

即 $|V(G)-S|\leq|S|\cdot\Delta(G)$．因此 $n-\gamma(G)\leq\gamma(G)\cdot\Delta(G)$，从而 $\gamma(G)\geq\left\lceil\frac{n}{1+\Delta(G)}\right\rceil$．

设 x 是 G 的顶点且 $\deg(x)=\Delta(G)$．则 $V(G)-N(x)$ 是基数为 $n-\Delta(G)$ 的一个支配集，所以 $\gamma(G)\leq n-\Delta(G)$．

定理 5　图 G 的一个顶点集 S 是一个独立支配集当且仅当 S 是一个极大独立集．

证明　因为每一个极大独立集是一个支配集．反之，假设 S 是一个独立支配集，则 S 是独立集且不属于 S 的每一个顶点都与 S 的一个顶点邻接，即 S 是极大独立集．

推论 2　图 G 的每个极大独立集是一个极小支配集．

证明　设 S 是图 G 的一个极大独立集．由定理 5，S 是一个支配集．因为 S 是独立集，显然 S 的每个顶点不与 S 中其他顶点邻接．这样，S 的每个顶点满足定理 1 中的性质(2)，所以，由定理 1，S 是极小支配集．

需要注意，不是每个支配集都是独立集，也不是每个最小支配集都是独立集．如图 5-9 所示，$S_1=\{1,3,6,8,10,12\}$ 是 G 的一个

最大独立集(自然是支配集);而 $S_2=\{4,5,9\}$ 是 G 的一个最小支配集且显然不是独立集;然后,$S_3=\{2,7,11\}$,既是最小支配集而且是独立集.

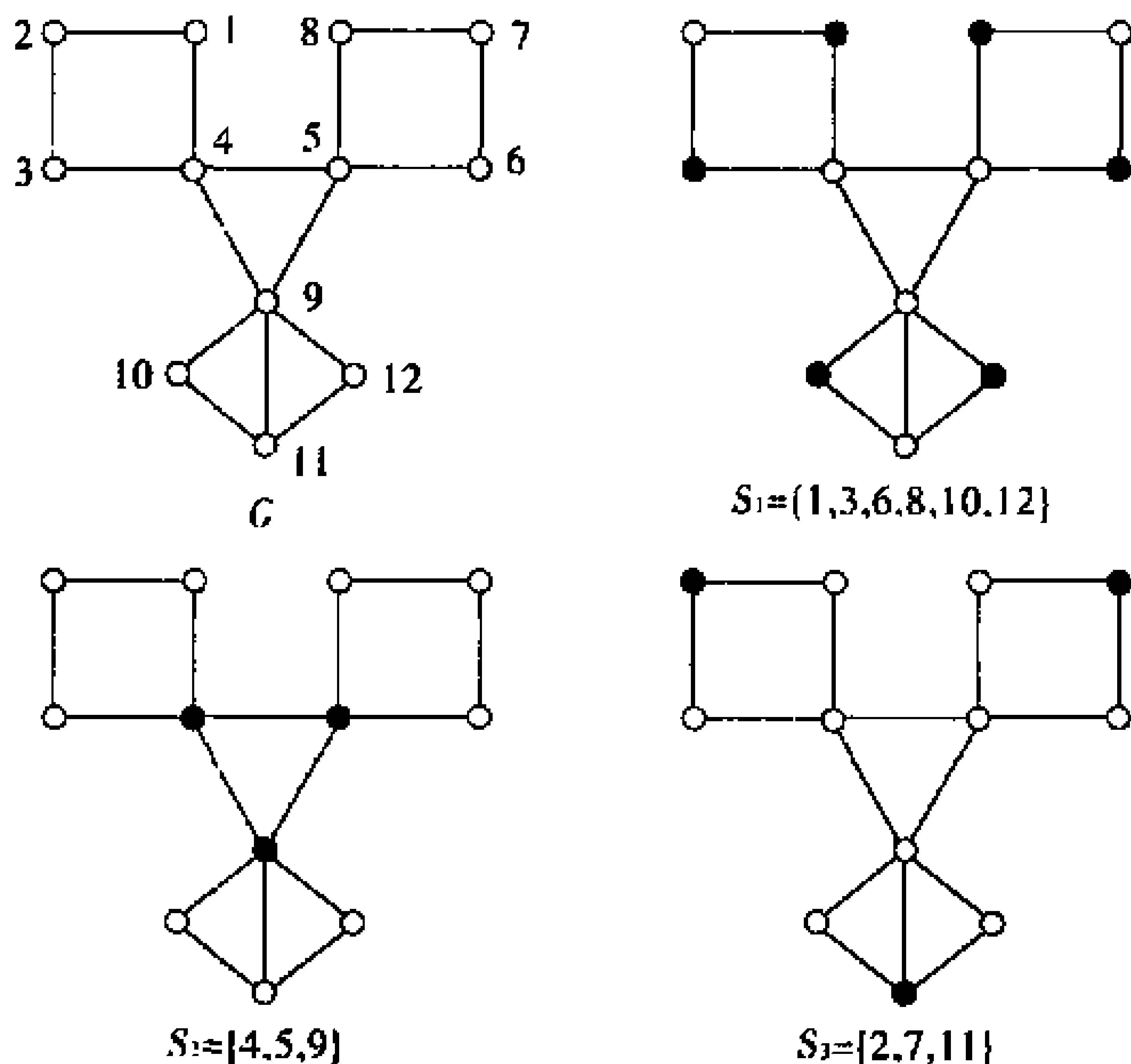


图 5-9

支配集也有其实际背景:要在 $V_1,V_2,\cdots,V_n$ 这 n 座城市中建立起一个通讯系统,为此,要从这 n 个城镇中选定几座城镇,在那里建立起通讯站,要求它们与其它各城镇相邻,同时,为减少造价,要使通讯站数目最少,有时还会增加其它要求,比如,在造价最低的条件下,需要造两套(或更多套)通讯系统,使得当一个系统出了故障时,可以启用另一套.

为解决这个问题,我们构作图 G:以城镇作为图 G 的顶点,仅

当两城之间有直通讯线路时，相应的两顶点连一边，则图 G 的最小支配集即为所求. 若建两套，则从图 G 的全部极小支配集 D_1，$D_2,\cdots,D_n$ 中选取 D_i 与 D_j，使得

$$D_i \cap D_j = \varnothing,$$

$$|D_i \cup D_j| = \min\{|D_p| + |D_q| \mid 1 \leq p+q \leq n, D_p \cap D_q = \varnothing\}.$$

下面给出找出图的所有极小支配集的方法.

对应于图 $G=\langle V,E\rangle$ 的每一个顶点 $v_i \in V$，有一布尔变量 v_i，以及布尔表达式：$\varphi_i = v_i + \sum\limits_{v_j \in adj(v_i)} v_j$

作
$$\Psi = \varphi_1 \varphi_2 \cdots \varphi_n = \prod_{v_i \in V} \varphi_i .$$

其中，表达式中"+"为逻辑加(布尔加)，积为逻辑积(布尔积).

例如，在图 5-10 所示图 G 中，

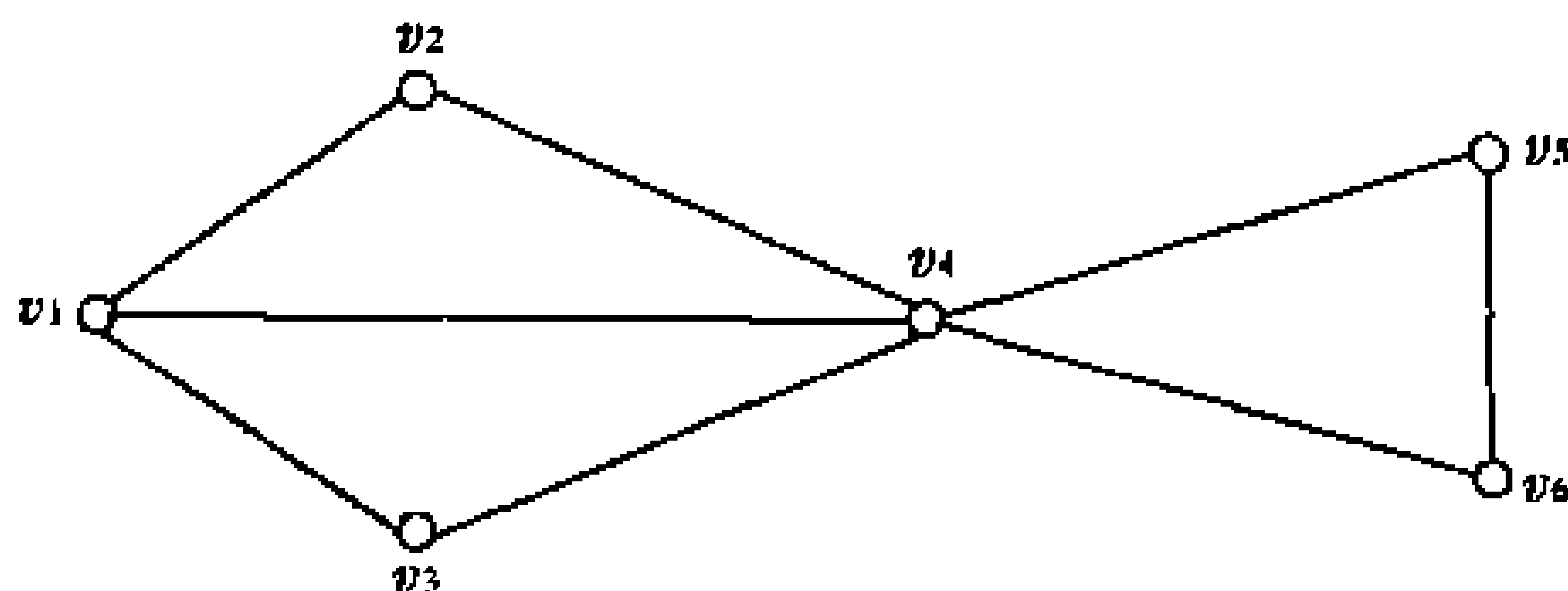


图 5-10

$$\varphi_1 = v_1 + v_2 + v_3 + v_4$$

$$\varphi_2 = v_2 + v_1 + v_3 + v_4$$

$$\varphi_3 = v_3 + v_1 + v_4$$

$$\varphi_4 = v_4 + v_1 + v_2 + v_3 + v_5 + v_6$$

$$\varphi_5 = v_5 + v_4 + v_6$$

$$\varphi_6 = v_6 + v_4 + v_5$$

而

$$\varphi_1\varphi_4=(v_1+v_2+v_3+v_4)(v_4+v_1+v_2+v_3+v_5+v_6)$$
$$=v_1+v_2+v_3+v_4$$
$$\varphi_2\varphi_3=(v_1+v_2+v_4)(v_1+v_3+v_4)$$
$$=(v_1+v_4)+(v_1+v_4)v_3+(v_1+v_4)v_2+v_2v_3$$
$$=v_1+v_4+v_2v_3$$
$$\varphi_5\varphi_6=(v_4+v_5+v_6)(v_4+v_5+v_6)$$
$$=v_4+v_5+v_6$$

所以 $\Psi=\varphi_1\varphi_2\varphi_3\varphi_4\varphi_5\varphi_6$

$$=(v_1+v_2+v_3+v_4)(v_1+v_4+v_2v_3)(v_4+v_5+v_6)$$
$$=(v_1+v_4+v_2v_3)(v_4+v_5+v_6)$$
$$=v_1v_5+v_1v_6+v_4+v_2v_3v_5+v_2v_3v_6$$

故得 G 的全部极小支配集如下：

$$\{v_1,v_5\},\{v_1,v_6\},\{v_4\},\{v_2,v_3,v_5\},\{v_2,v_3,v_6\}$$

第四节　匹配

匹配问题是运筹学的重要问题之一，也是图论的重要内容，它在所谓"人员分配问题"和"最优分配问题中"有重要作用.

定义 1　设 G 是无环图，$M\subseteq E(G)$，$M\neq\emptyset$，如果 M 中任意两条边在 G 中均不相邻，则称 M 是图 G 的一个**匹配(matching)**.若对图 G 的任何匹配 M'，均有 $|M'|<|M|$，则称 M 为图 G 的**最大匹配(greatest matching)**. G 中最大匹配中的边数称为**匹配数(matching number)**，记作 $\alpha'(G)$.

定义 2　设 M 是图 G 的匹配，G 中与 M 中的边关联的顶点称为 M **饱和点(M－Saturates)**，否则称为非 M **饱和点**.若图 G 的顶点都是 M 饱和点，则称 M 为 G 的**完美匹配(perfect matching)**.

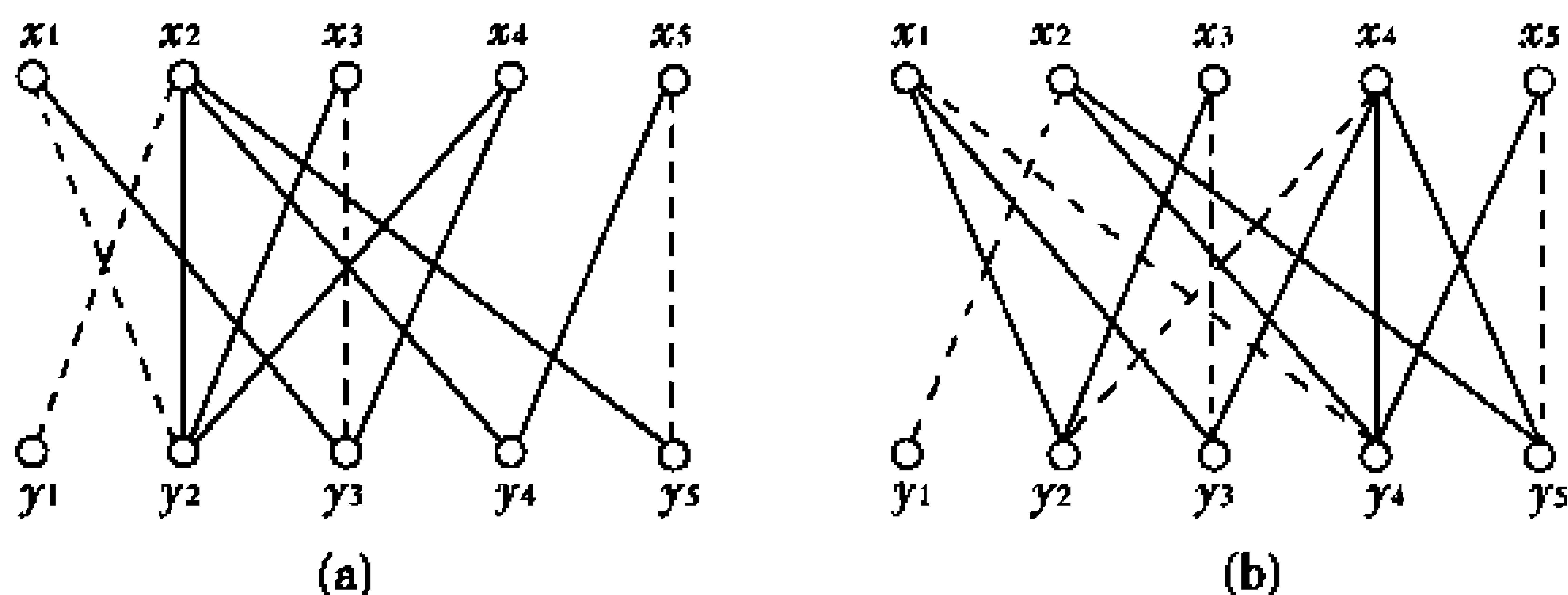


图 5-11

注释(1)完美匹配是最大匹配,反之未必.

(2)匹配的概念与边的方向无关,故只须讨论无向图.

(3)图 G 的边不交匹配的最小数目即为图 G 的边色数.

例如,图 5-11(a)中虚线所示为最大匹配,图 5-11(b)中虚线所示为完美匹配.

定义 3 设 M 是图 G 的匹配,P 是 G 的一条路,且在 P 中,M 的边和 $E(G)-M$ 的边交替出现,则称 P 是 G 的一条 M **交错路(M-alternating path)**.若 M 交错路 P 的两个端点为 M 非饱和点,则称 P 为 M **可增广路(M-augmenting path)**.

例如,图 5-12 中虚线所示为匹配 M,则(2,4,5,7,9,10)是一条 M 交错路,而(1,2,4,5,7,8)是一条 M 可增广路.

由定义可知,M 可增广路 P 一定包含奇数条边,且 P 中不属于 M 的边比属于 M 中的边多一条.为什么把起点与终点是非饱和点的 M 交错路称为 M 可增广路呢?这是因为一旦在 G 中找到一条这样的 M 可增广路 P,就可以对现有的匹配 M 进行调整,而得到另一个比 M 多一条边的匹配 M'.调整的方法如下:把可增广路 P 上原来在匹配 M 上的边从匹配 M 中划去,而把 P 上原来不在 M 中的边加到 M 中去,得到 G 的一个边子集 $M'=(M-E(P))\cup(E(P)-M)=M\oplus E(P)$.由于 M 可增广路 P 的起点与终点是 M 非饱和点,用上面方法调整后所得的 M' 仍是 G 的一个

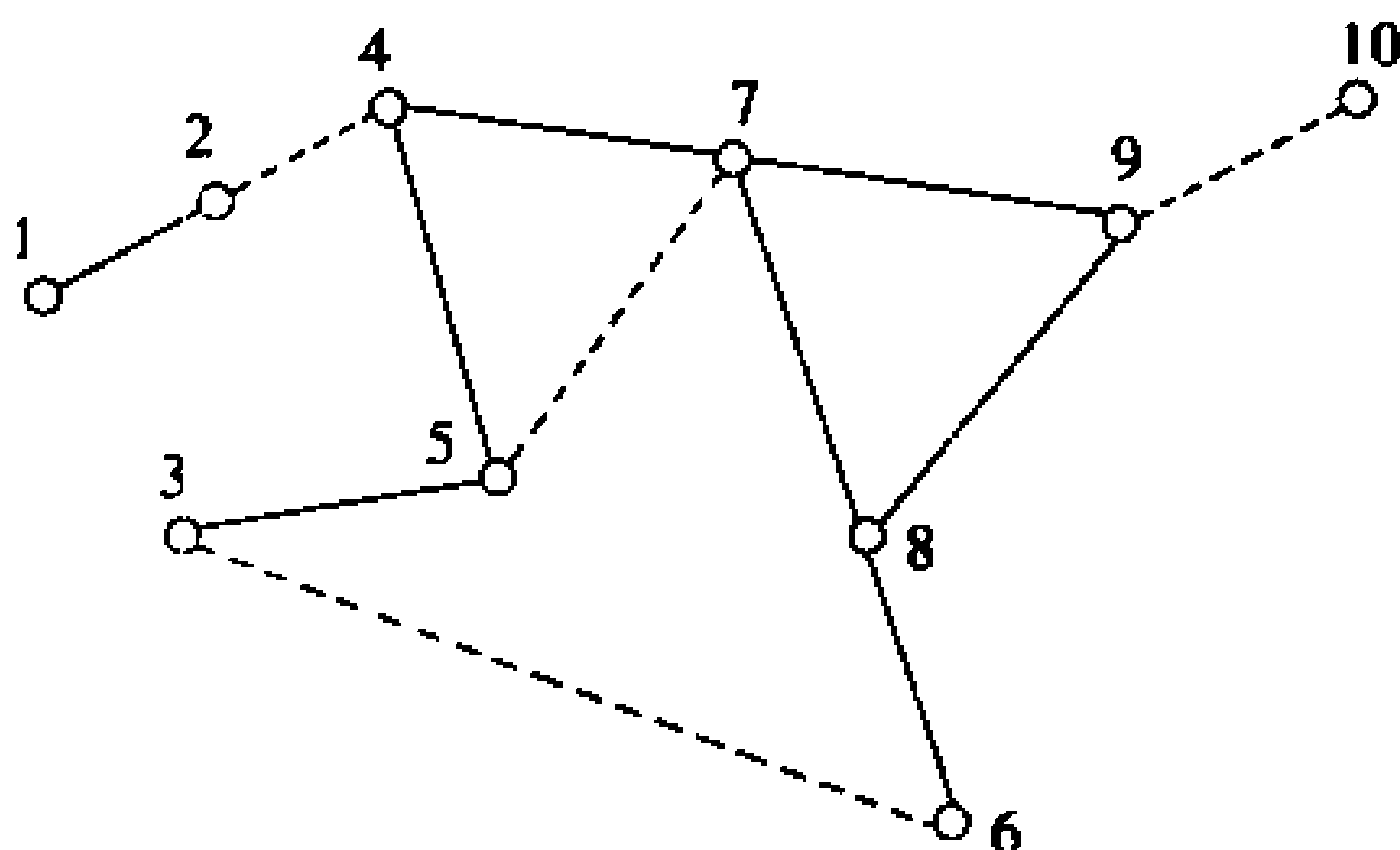


图 5-12

匹配，且 $|M'|=|P\oplus M|=|M|+1$.

定理 1　设 M_1 和 M_2 是图 G 的两个不同匹配，由 $M_1\oplus M_2$ 导出的 G 的边导出子图记作 H，则 H 的任意连通分支是下列情况之一：

(1)边在 M_1 和 M_2 中交错出现的偶圈.

(2)边在 M_1 和 M_2 中交错出现的路.

(3)孤立结点.

证明　$H=M_1\oplus M_2$，显然 $\delta(H)\geqslant 1$. 由于 M_1 和 M_2 是 G 的匹配，所以 H 的任意顶点 x 至多与一条 M_1 的边关联，同时也至多与一条 M_2 的边关联，即 $\deg(x)\leqslant 2$，由此可知 $\Delta(H)\leqslant 2$，故 H 的每个分支或者是一条路或者是一个圈. 由匹配的定义知，H 的任意两条邻接边一定分别属于不同的匹配 M_1 和 M_2，从而每条路或者圈的边交错地属于 M_1 和 M_2 且每个圈都是偶圈. 孤立结点即为关联 $M_1\cap M_2$ 中边的顶点. 证毕.

定理 2　M 是图 G 的最大匹配，当且仅当 G 中不存在 M 可增广路.

证明　($\Leftarrow$)若存在 M 可增广路 P，则 $M'=M\oplus P$ 是 G 的一个新匹配，且 $|M'|=|M|+1>|M|$，这是不可能的.

(⇒)假设 M 不是 G 的最大匹配，则存在一个匹配 M'，使得 $|M'|>|M|$．作 $H=M'\oplus M$，由定理 1 知，H 的任意连通分支 Q 必是下列三种类型之一：

(1)孤立结点．Q 中的每个结点度数均为 0，该结点即为关联 $M'\cap M$ 中边的顶点．

(2)交错偶圈．Q 中的每个结点度数均为 2．

(3)交错路．Q 中含有 1 度点 x，且必含另一个 1 度点 y（奇度点有偶数个），所以 Q 是以 x 和 y 为端点的路．Q 中内部点（若存在的话）都是 2 度点．

由于 $|M'|>|M|$，所以 $|E(H)\cap M'|>|E(H)\cap M|$，因而 H 中必有一条起始于 M' 终止于 M' 的连通分支 P，于是 P 是 M 可增广路．证毕

设 S 是图 G 的任一顶点子集，G 中与 S 的顶点邻接的所有顶点的集合，称为 S 的**邻集(neighbour set)**，记作 $N_G(S)$．

定理 3　(Hall 定理，1935)设 G 是有二部划分 (V_1,V_2) 的二分图，则 G 含有饱和 V_1 的每个顶点的匹配 M 的充要条件是，对 $\forall S\subseteq V_1$，$|N(S)|\geqslant|S|$．

证明　(⇒)对 $\forall S\subseteq V_1$，匹配 M 将 S 中的每个顶点与 $N(S)$ 中顶点配对，故 $|N(S)|\geqslant|S|$．

(⇐)$\forall S\subseteq V_1$，有 $|N(S)|\geqslant|S|$．可以按下面方法作出饱和 V_1 的匹配 M．

先作任一初始匹配 M_1，若已饱和 V_1，定理得证．否则，V_1 中至少有一个非饱和点 x_1，检查以 x_1 为起点，终点在 V_2 中的交错路．考虑下列两种情形：

(1)不存在任何一条交错路可以到达 V_2 的非饱和点．这时从 x_1 开始的一切交错路的终点还是在 V_1．故存在 $A\subseteq V_1$，使 $|N(A)|<|A|$，这与已知矛盾．

(2)存在一条以 x_1 为起点，终点为 V_2 的非饱和点的交错路 P，可见 P 是可增广路，于是作新匹配 $M_2=M_1\oplus P$，显然 M_2 饱和 x_1，且 $|M_2|>|M_1|$.

因此，重复以上过程，可以找到饱和 V_1 的全部顶点的匹配 M. 定理的充分性得证.

推论 1 具有二部划分(V_1,V_2)的二分图 G 有完美匹配$\Leftrightarrow$ $|V_1|=|V_2|$，且对 $\forall S\subseteq V_1$(或 V_2)均有 $|N(S)|\geqslant|S|$.

推论 2 设 G 是 $k(>0)$正则二分图，则 G 有完美匹配.

证明 证 G 是二部划分(V_1,V_2)的 k 正则二分图，则

$$k|V_1|=|E(G)|=k|V_2|$$

由于 $k\neq 0$，所以 $|V_1|=|V_2|$. 任取 $S\subseteq V_1$，并用 E_1 和 E_2 分别表示 G 中与 S 和 $N(S)$ 中点关联的边集，则 $E_1\subseteq E_2$.

因而 $k|N(S)|=|E_2|\geqslant|E_1|=k|S|$

即 $|N(S)|\geqslant|S|,\forall S\subseteq V_1$

由定理 3 可知，G 有饱和 V_1 的匹配 M. 由于 $|V_1|=|V_2|$，所以 M 是完美匹配.

推论 3 设 G 是二部划分(V_1,V_2)的简单二分图，且 $|V_1|=|V_2|=n$，若 $\delta(G)\geqslant\frac{n}{2}$，则 G 有完美匹配.

证明 任取 $S\subseteq V_1$. 若 $|N(S)|<|S|$，由于 G 是简单二分图，且 $\delta(G)\geqslant\frac{n}{2}$，所以 $|S|>|N(S)|\geqslant\delta(G)\geqslant\frac{n}{2}$，且 $V_2-N(S)\neq\varnothing$. 令 $u\in V_2-N(S)$，则 $N(u)\subseteq V_1-S$，即

$\delta(G)\leqslant\deg(u)=|N(u)|\leqslant|V_1|-|S|<n-\frac{n}{2}=\frac{n}{2}$，这与 $\delta(G)\geqslant\frac{n}{2}$ 矛盾. 故 $|N(S)|\geqslant|S|,\forall S\subseteq V_1$，故 G 有完美匹配.

定理 4 G 有完美匹配$\Leftrightarrow O(G-S)\leqslant|S|,\forall S\subset V(G)$，其

中$O(G-S)$是 $G-S$ 的奇阶连通分支数目.

证明 显然只须对简单图证明这个定理.

($\Rightarrow$)设 M 是 G 的一个完美匹配,并设 $S\subset V(G)$,$G_1,G_2,\cdots,G_n$ 是$G-S$ 的奇阶通分支.所以存在 $x_i\in V(G_i)$和 $y_i\in S$ 使得$\{\{x_i,y_i\}\mid i=1,2,\cdots,n\}\subseteq M$.于是

$$O(G-S)=n=|\{y_1,y_2,\cdots,y_n\}|\leqslant|S|.$$

($\Leftarrow$)当 $S=\emptyset$ 时,$O(G-\emptyset)=O(G)\leqslant 0$,从而 G 只有偶分支且阶P 为偶数,由此可得对 $V(G)$的任意子集 S,$O(G-S)$与$|S|$有相同的奇偶性.

对正偶数 P 进行归纳.当 $P=2$ 时,如果 G 是P 阶图使得对$V(G)$的任意子集 S,$O(G-S)\leqslant|S|$,则 $G\simeq K_2$,结论成立.

假设结论对任何$<P$(P 是不小于 4 的偶数)阶的图成立,并设 G 是P($\geqslant 4$,偶)阶图.设 T 是$V(G)$中满足定理已知条件的最大非空子集.令$|T|=m$,并令 $G_1,G_2,\cdots,G_m$ 是$G-T$ 的奇阶分支,则有下列三个结论:

(1)$G-T$ 无偶阶分支.事实上,设 H 是$G-T$ 的一个偶阶分支,$u\in V(H)$,则 $m+1\leqslant O(G-(T\cup\{u\}))\leqslant|T\cup\{u\}|=m+1$,

即 $O(G-(T\cup\{u\}))=|T\cup\{u\}|$.这与 T 的选取矛盾.

(2)任取 $x\in G_i$,则 G_i-x 有完美匹配.否则,由归纳假设存在 $S\subset V(G_i-x)$使得 $O((G_i-x)-S)>|S|$.由于 $O((G_i-x)-S)$与$|S|$有相同的奇偶性.所以

$$O((G_i-x)-S)\geqslant|S|+2$$

于是

$$\begin{aligned}|T|+1+|S|=|T\cup S\cup\{x\}|&\geqslant O(G-(T\cup S\cup\{x\}))\\&=O(G-T)-1+O((G_i-x)-S)\\&\geqslant|T|+1+|S|\end{aligned}$$

即 $O(G-(T\cup S\cup\{x\}))=|T\cup S\cup\{x\}|>|T|$，这与 T 的选取矛盾.

(3) G 含匹配 $M=\{\{s_i,t_i\}\mid s_i\in T,t_i\in V(G_i),i=1,2,\cdots,m\}$. 事实上，考虑二部划分为 (V_1,V_2) 的二分图 H，其中 $V_1=\{G_1,\cdots,G_m\}$，$V_2=T$，G_i 与 T 中点 y 在 H 中相邻 $\Leftrightarrow G$ 含从 y 到 G_i 中点的边. 于是(3)成立 $\Leftrightarrow H$ 有饱和 V_1 的匹配. 任取 $A\subseteq V_1$，并令 $B=N_H(A)\subseteq V_2$，则由于 A 中元素都是 $G-B$ 的奇阶分支，有

$$|A|\leqslant O(G-B)\leqslant|B|=|N_H(A)|$$

由定理 3 知，于是(3)成立.

综合(1)，(2)和(3)，定理得证.

例 1 有 n 张纸牌，每张纸牌的正反两面都写上 $1,2,\cdots,n$ 的某一个数. 证明：如果每个数字都恰好出现两次，那么这些纸牌一定可以这样摊开，使朝上的面中 $1,2,\cdots,n$ 都出现.

解 作一个二分图 $G=\langle V_1,V_2,E\rangle$，其中 $V_1=\{1,2,\cdots,n\}$，$V_2=\{y_1,y_2,\cdots,y_n\}$ 表示这 n 张纸牌. i 与 y_j 之间连接的边数等于数 i 在纸牌 y_j 中出现的次数，这样得到的图 G 是一个 2 正则二分图，因此 G 中有完美匹配，设为 $M=\{1y_{i_1},2y_{i_2},\cdots,ny_{i_n}\}$，则只要把纸牌 y_{i_1} 中的 1 朝上，y_{i_2} 中的 2 朝上，……，y_{i_n} 中的 n 朝上，这样摊开的纸牌就能使朝上面中 $1,2,\cdots,n$ 都出现.

例 2 某工厂生产由 6 种不同颜色的纱织成的双色布，由这个工厂所生产的双色布中，每一种颜色至少和其他三种颜色搭配. 证明可以挑选出三种不同的双色布，它们含有所有的 6 种颜色.

证明 构造图 $G=\langle V,E\rangle$，其中 $V=\{v_1,v_2,v_3,v_4,v_5,v_6\}$ 表示 6 种颜色，工厂生产出一种由颜色 v_i 与 v_j 搭配而成的双色布 $\Leftrightarrow$ 边 $\{v_i,v_j\}\in E(G)$. 由题意知，G 为简单图，且每个结点的度

数至少为 3，每条边对应一种双色布. 下面只要证明图 G 含有一个完美匹配.

不妨设边 $\{v_1, v_2\} \in E(G)$，由于 $d(v_3) \geqslant 3$，存在一个不同于 v_1 和 v_2 的顶点 $v_i (4 \leqslant i \leqslant 6)$，使 $\{v_3, v_i\} \in E(G)$，不妨设 $v_i = 4$，即 $\{v_3, v_4\} \in E(G)$.

如果边 $\{v_5, v_6\} \in E(G)$，则 $\{v_1, v_2\}$，$\{v_3, v_4\}$，$\{v_5, v_6\}$ 就是 G 的一个完美匹配.

如果边 $\{v_5, v_6\} \notin E(G)$，由于 $d(v_5) \geqslant 3$，v_1, v_2, v_3, v_4 中至少有 3 个顶点与 v_5 相邻，即 v_5 与边 $\{v_1, v_2\}$，$\{v_3, v_4\}$ 中的每一边的某一个端点相邻，不妨设 $\{v_1, v_5\} \in E(G)$ 和 $\{v_3, v_5\} \in E(G)$.

对于顶点 v_6，同样与 v_1, v_2, v_3, v_4 中的至少 3 个点相邻，即在 v_2 和 v_4 中至少有 1 个点与 v_6 相邻. 如果 $\{v_2, v_6\} \in E(G)$，则边 $\{v_1, v_5\}$，$\{v_3, v_4\}$，$\{v_2, v_6\}$ 是 G 的一个完美匹配；如果 $\{v_4, v_6\} \in E(G)$，则 $\{v_1, v_2\}$，$\{v_3, v_5\}$，$\{v_4, v_6\}$ 是 G 的一个完美匹配.

综上所述，G 总存在完美匹配，完美匹配中的三条边所对应的三种双色布即为所求.

第五节　最大匹配的生成算法

在匹配的应用问题中，我们常常需要求出给定图的最大匹配. 本节给出一个有效算法，它是由匈牙利数学家 Egervary(1931) 首先提出来的，故通常称为**匈牙利算法**.

设 M 是图 G 的匹配，x 是 G 中非 M 饱和点. G 中由起点为 x 的 M 交错路所能连接的顶点集所导出的 G 的导出子图称为**根在 x 的 M 交错子图(alternating subgraph rooted at x)**.

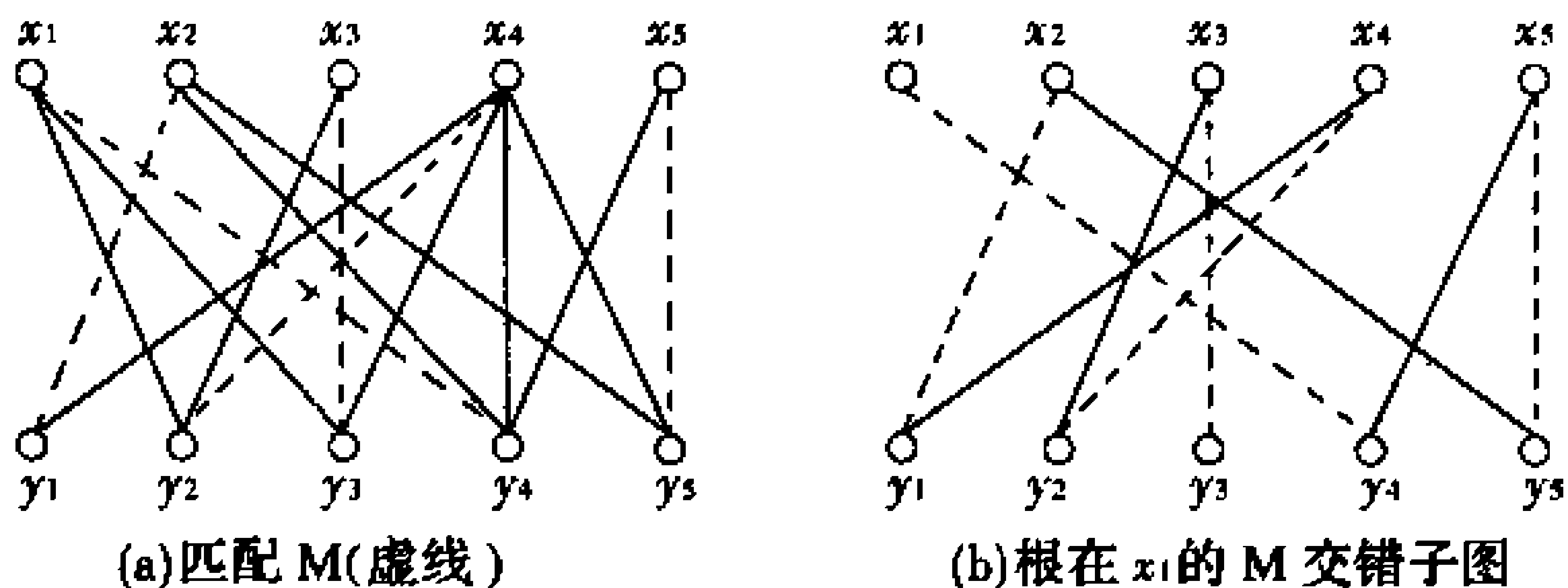


图 5-13

如图 5-13(a)所示，已知图 G 的匹配 M，以 x_1 为起点的 M 交错路 $p=(x_1y_4x_5y_5x_2y_1x_4y_2x_3y_3)$，由 $V(P)$ 导出的根在 x_1 的 M 交错子图如图 5-13(b)所示.

定理 1　设 M 是具有二部划分 (V_1,V_2) 的二分图 G 的匹配，$x\in V_1$ 是非 M 饱和点，H 是 G 中根在 x 的 M 交错子图的顶点集，$S=H\cap V_1$，$T=H\cap V_2$，则

(1) $T\subseteq N_G(S)$

(2)下述三条等价：

(a) G 中不存在以 x 为端点的 M 可增广路；

(b) x 是 H 中唯一的非 M 饱和点；

(c) $T=N_G(S)$，且 $|T|=|S|-1$

证明　(1) $\forall y\in T$，则 G 中存在以 x 和 y 为端点的 M 交错路 P. 令 $u\in N_p(y)$，由于 G 是二分图且 $y\in T\subseteq V_2$，所以 $u\in H\cap V_1=S$，即 $y\in N_G(S)$，因而 $T\subseteq N_G(S)$

(2) $(a)\Rightarrow(b)$ 设 y 是 H 中异于 x 的非 M 饱和点，则 G 中存在以 x 和 y 为端点的 M 交错路 P. P 是 G 中以 x 为端点的 M 可增广路，这与 (a) 矛盾.

$(b)\Rightarrow(a)$ 设 G 中存在以 x 为端点的 M 可增广路 P，并设 P 的另一端点为 $y(\neq x)$，则 y 是非 M 饱和点. 由 H 的定义知，$y\in$

H,这与(b)矛盾.

$(b)\Rightarrow(c)$任取 $y\in N_G(S)\subseteq V_2$,于是存在 $u\in S=H\cap V_1$ 和边 $e\in E(G)$使 $\varphi_G(e)=\{u,y\}$.若 $u=x$,则显然有 $y\in T$.下设 $u\neq x$,于是 G 中存在以 x 和 u 为端点的 M 交错路 P.由于 x 是非 M 饱和点,所以 u 为 M 饱和点.若 P 不含 y,则 $e\notin M$.由 H 的定义知,$y\in H\cap V_2=T$,因而有 $N_G(S)\subseteq T$,再由(1)知,$T=N_G(S)$.

由于 x 是 H 中唯一的非 M 饱和点,所以 T 中点全是 M 饱和点.又由于 V_1 中通过 M 与 T 中点配对的点全在 S 中,且 $T=N_G(S)$,所以 $S-\{x\}$ 中点与 T 中点由 M 配对.故 $|T|=|S|-1$.

$(c)\Rightarrow(b)$任取 $u\in S-\{x\}$.设 P 是 G 中以 x 和 u 为端点的 M 交错路.由于 G 是二分图,并且 $x,u\in V_1$,所以 P 的长为偶数.又由于 x 是非 M 饱和点,所以 u 是 M 饱和点.由 $u\in S-\{x\}$ 的任意性知,$S-\{x\}$ 中点全是 M 饱和点,它们与 $N_G(S)$ 中点由 M 配成对.由于 $N_G(S)=T$ 且 $|T|=|S|-1$,所以 T 中点全是 M 饱和点,即知 x 是 H 中唯一非 M 饱和点.

匈牙利算法的基本思想较简单.设 G 是具有二部划分(V_1,V_2)的二分图,从图 G 的任意匹配 M 开始.若 M 饱和 V_1,则 M 是 G 的最大匹配.若 M 不能饱和 V_1,则在 V_1 中选择一个非 M 饱和点 x.若 G 中存在以 x 为起点的 M 可增广路 P,则 $M'=M\oplus P$ 就是比 M 更大的匹配,利用 M' 代替 M,并重复这个过程.若 G 中不存在以 x 为起点的 M 可增广路,则令 H 是根在 x 的 M 交错子图的顶点集,并令 $S=H\cap V_1$,$T=H\cap V_2$.由定理 1 可知,$N_G(S)=T$,且 G 中不存在以 x 为起点的 M 可增广路,此时称 x 为检验过的非 M 饱和点.对 V_1 中其它未检验过的非 M 饱和点重复这个过程,直到 V_1 中的所有非 M 饱和点全部检验过为止.

当整个过程结束时，由于 G 中不存在 M 可增广路，从而 M 为 G 的最大匹配.

匈牙利算法

设 G 是具有二部划分 (V_1,V_2) 的二分图.

(1)任给初始匹配 M；

(2)若 M 饱和 V_1 则结束，否则转(3)；

(3)在 V_1 中找一非 M 饱和点 x，置 $S=\{x\}$，$T=\varnothing$；

(4)若 $N(S)=T$，则停止，否则任选一点 $y\in N(S)-T$；

(5)若 y 为 M 饱和点转(6)，否则作

求一条从 x 到 y 的 M 可增广路 P，置 $M=M\oplus P$，转(2)

(6)由于 y 是 M 饱和点，故 M 中有一边 $\{y,u\}$，置 $S=S\cup$

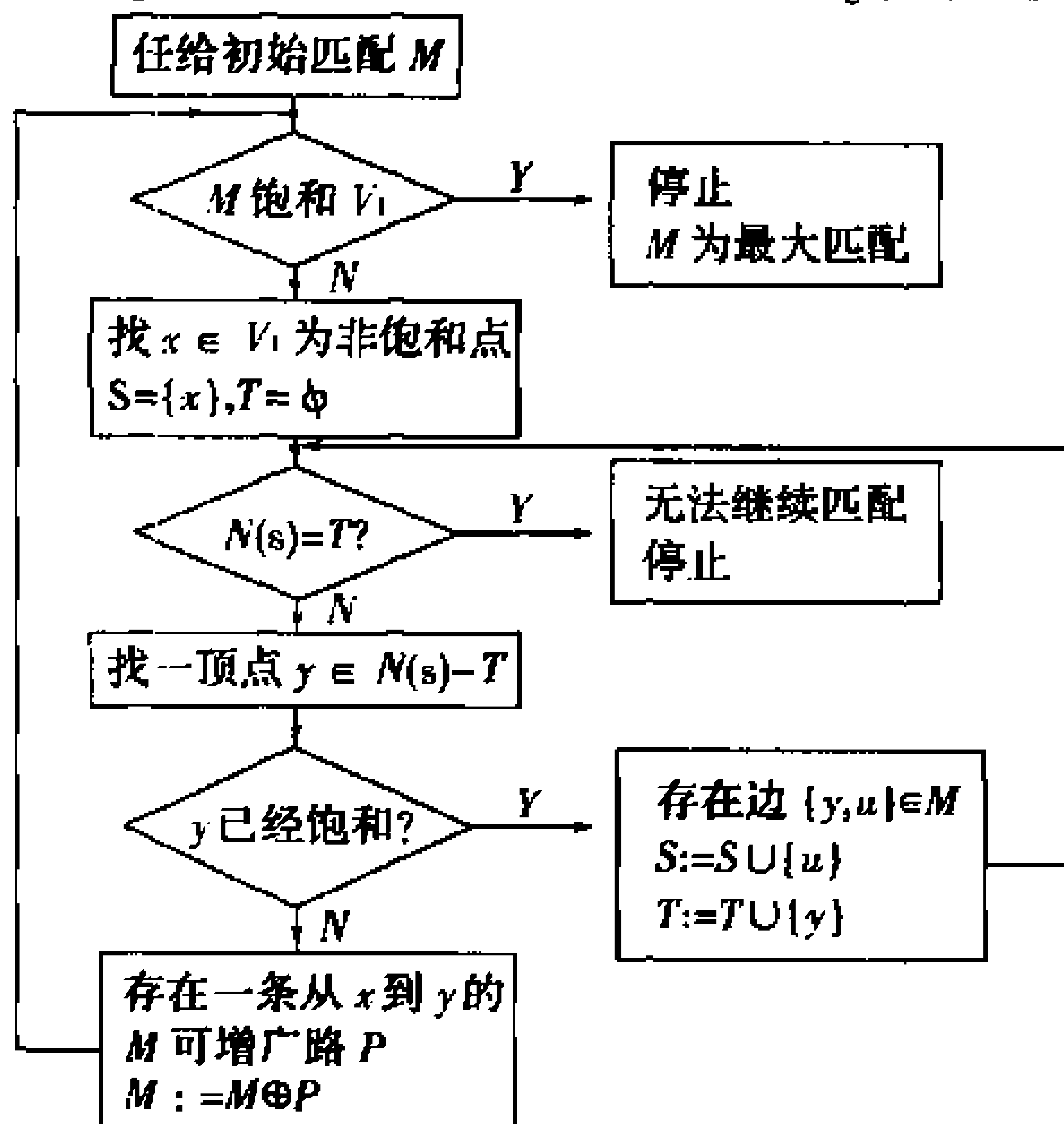


图 5-14

$\{u\}$, $T=T\cup\{y\}$,转(4).

匈牙利算法的流程图如图 5-14 所示.

例 1 如图 5-15(a)所示,已知二部划分(V_1, V_2)的二分图 G,其中 $V_1=\{x_1,x_2,x_3,x_4,x_5\}$, $V_2=\{y_1,y_2,y_3,y_4,y_5\}$,试求图 G 的最大匹配.

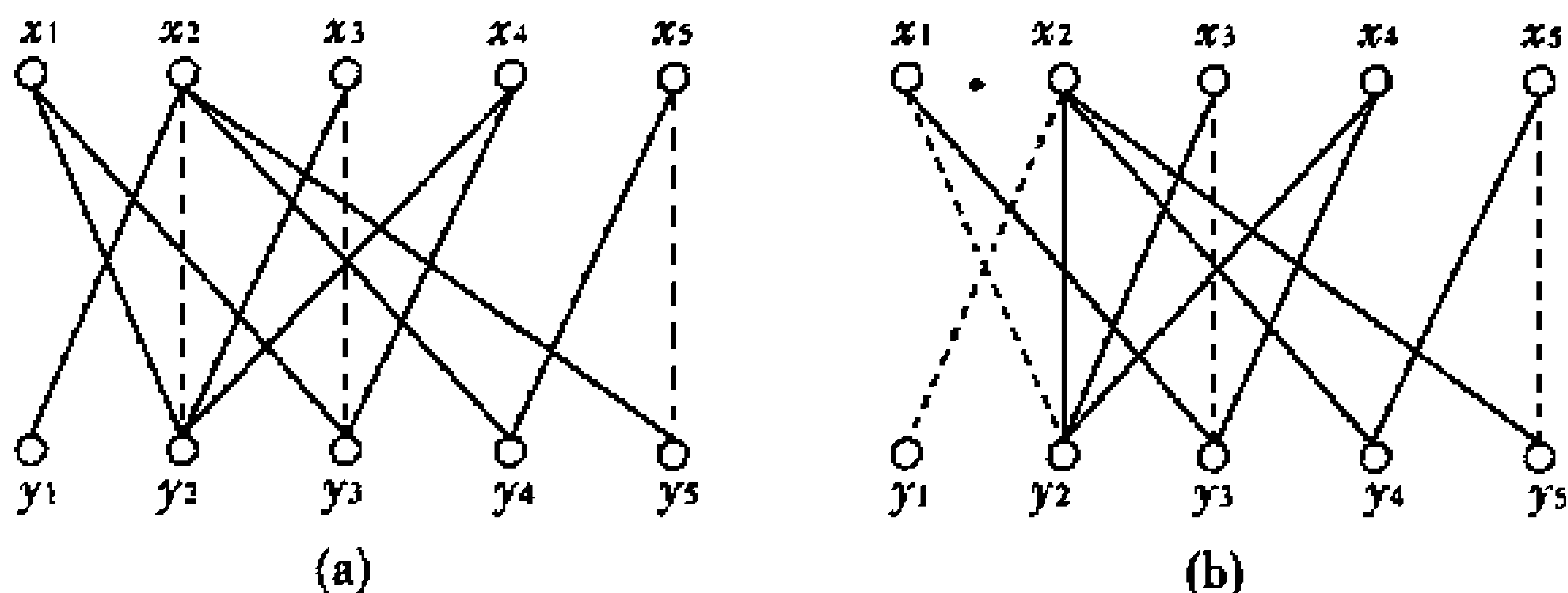


图 5-15

解 任取初始匹配 $M=\{x_2y_2, x_3y_3, x_5y_5\}$,如图 5-15(a)中虚线所示.解题过程如表 5-3 所示:

表 5-3

M	x	S	T	$N(S)$	$y\in N(S)-T$	$\{y,u\}\in M$	P
$\{x_2y_2, x_3y_3, x_5y_5\}$	x_1	$\{x_1\}$	$\varnothing$	$\{y_2,y_3\}$	y_2 饱和	$\{y_2,x_2\}$	
		$\{x_1,x_2\}$	$\{y_2\}$	$\{y_1,y_2,y_3,y_4,y_5\}$	y_1 非饱和		$(x_1y_2x_2y_2)$
$\{x_1y_2, x_2y_1, x_3y_3, x_5y_5\}$	x_4	$\{x_4\}$	$\varnothing$	$\{y_2,y_3\}$	y_2 饱和	$\{y_2,x_1\}$	
		$\{x_4,x_1\}$	$\{y_2\}$	$\{y_2,y_3\}$	y_3 饱和	$\{y_3,x_3\}$	
		$\{x_4,x_1,x_3\}$	$\{y_2,y_3\}$	$\{y_2,y_3\}$	$N(S)=T$,停止		

因此，$M=\{x_1y_2, x_2y_1, x_3y_3, x_5y_5\}$即为图 G 的最大匹配，如图 5-15(b)中虚线所示.

设二分图 G 有 n 个结点 m 条边，运用匈牙利算法求 G 的最大匹配时，初始匹配可以空，因此，算法最多找 n 条可增广路，每找一条可增广路，最多比较 m 条边，因而算法的时间复杂度为 $O(nm)$. 故匈牙利算法为有效算法.

第六节　最优匹配

某公司准备安排 n 个职员 $x_1, x_2, \cdots, x_n$ 从事 n 项工作 $y_1, y_2, \cdots, y_n$. 已知每个职员能胜任其中一项或几项工作，试问怎样安排，才能使尽量多的人有工作可做，同时使尽量多的工作有人胜任?

构作具有二部划分(V_1, V_2)的简单二分图 G，其中 $V_1=\{x_1, x_2, \cdots, x_n\}$，$V_2=\{y_1, y_2, \cdots, y_n\}$，并且边$\{x_i, y_j\}\in E(G)\Leftrightarrow$职员 x_i 胜任工作 y_j，于是问题转化为求给定二分图 G 的最大匹配.

我们知道，这种分配方案可能不止一种，或者说职员做各项工作，熟练程度、工作效率等未必一致，因此，要制定一个分工方案，使人尽其才，且公司的总效益最大. 这样，就要考察具有二部划分(V_1, V_2)的加权完全二分图$(K_{n,n}, w)$，其中 $V_1=\{x_1, x_2, \cdots, x_n\}$，$V_2=\{y_1, y_2, \cdots, y_n\}$，边$\{x_i, y_j\}$上的权 $w(x_i, y_j)$表示职员 x_i 做工作 y_j 的效率. 于是，问题等价于在这个加权图$(K_{n,n}, w)$中求一个总权最大的完美匹配，我们称这种匹配为**最优匹配**(**optimal matching**).

当然，若枚举所有 $n!$ 个完美匹配，然后比较它们的权，这种方法无疑是可以的. 但是，当 n 很大时，这种方法显然是无效的.

本节将介绍一个求最优匹配的有效算法.

定义 1 已知 G 是具有二部划分(V_1,V_2)的完全加权二分图,映射 $l:V(G)\to R$,满足对 G 的每条边 $e=\{x,y\}$,均有 $l(x)+l(y)\geqslant w(x,y)$,其中 $w(x,y)$表示边$\{x,y\}$的权,则称 l 为 G 的**可行顶标(feasible vertex labelling)**.令 $E_l=\{\{x,y\}\mid\{x,y\}\in E(G),l(x)+l(y)=w(x,y)\}$,$G_l$ 为以E_l 为边集的 G 的生成子图,则称 G_l 为 l **等子图(equality subgraph)**.

可行顶标总是存在的,例如.

$$\begin{cases} l(x)=\max\limits_{y\in V_2} \quad w(x,y)\in V_1 \\ l(y)=0 \qquad\qquad y\in V_2 \end{cases}$$

这种可行顶标称为**平凡标号(trivial labelling)**.

定理 1 设 l 是 G 的可行顶标.若 l 等子图 G_l 有完美匹配 M,则 M 是 G 的最优匹配.

证明 由于 G_l 是 G 的生成子图,M 是 G_l 的完美匹配,所以,M 也是 G 的完美匹配.又由于对于匹配 M 的每条边 e,都有 $e\in E(G_l)$,而且 M 中每条边覆盖每个顶点正好一次.所以

$$w(M)=\sum_{e\in M}w(e)=\sum_{x\in V_1}l(x)$$

另一方面,对于 G 的任何完美匹配 M' 有

$$w(M')=\sum_{e\in M'}w(e)\leqslant\sum_{x\in V_1}l(x)$$

于是 $w(M)\geqslant w(M')$,即 M 是 G 的最优匹配.

基于上述定理,kuhn(1955)和 Munkres(1957)提出一个在加权完全二分图$(K_{n,n},w)$中求最优匹配的有效算法.算法的基本思想是:

首先给出 $K_{n,n}$任意的可行顶标(如平凡标号),然后决定 G_l.在 G_l 中执行匈牙利算法.若在 G_l 中找到完美匹配,则这个完美

匹配就是 G 的最优匹配.否则,匈牙利算法终止于 $S\subset V_1$,$T\subset V_2$,且 $N_{G_l}(S)=T$.令

$$a_l=\min\{l(x)+l(y)-w(x,y)\mid x\in s,y\in V_2-T\}$$

则由

$$l'(u)=\begin{cases}l(u)-a_l & u\in s\\ l(u)+a_l & u\in T\\ l(u) & \text{其它}\end{cases}$$

确定了一个新的可行顶标 l'.此时 $a_l>0$,且 $T\subset N_{G_{l'}}(S)$.以 l'替代 l.连续进行这种修改,直到存在一个等子图含完美匹配为止.由于最优匹配必存在,所以这种修改必在有限步结束.

kuhn - munkres 算法

(1)从任意可行顶标(如平凡标号)l 开始,确定 l 等子图 G_l,并且在 G_l 中选取匹配 M.若 M 饱和 V_1,则 M 是完美匹配,也即 M 是最优匹配,算法终止.否则转入(2)步.

(2)匈牙利算法终止于 $S\subset V_1$,$T\subset V_2$ 且使 $N_{G_l}(S)=T$.计算 a_l,确定新的可行顶标 l',并以 l'替代 l,以 $G_{l'}$ 替代 G_l 转入(1)步.

例 1 已知完全二分图 $K_{5,5}$,其中 $V_1=\{x_1,x_2,x_3,x_4,x_5\}$,$V_2=\{y_1,y_2,y_3,y_4,y_5\}$,且 $K_{5,5}$的权矩阵为 A,求 $K_{5,5}$的最优匹配.

$$A=\begin{bmatrix}3&5&5&4&1\\2&2&0&2&2\\2&4&4&1&0\\0&1&1&0&0\\1&2&1&3&3\end{bmatrix}$$

解 (1)取可行顶标 l 如下:

$$l(y_1)=l(y_2)=l(y_3)=l(y_4)=l(y_5)=0$$
$$l(x_1)=\max(3,5,5,4,1)=5$$
$$l(x_2)=\max(2,2,0,2,2)=2$$
$$l(x_3)=\max(2,4,4,1,0)=4$$
$$l(x_4)=\max(0,1,1,0,0)=1$$
$$l(x_5)=\max(1,2,1,3,3)=3$$

(2)取 G_l 及 G_l 的匹配(虚线)如图 5-16 所示. 由于 $O(G-x_2)=3$,故 G_l 中无完美匹配,需修改顶标.

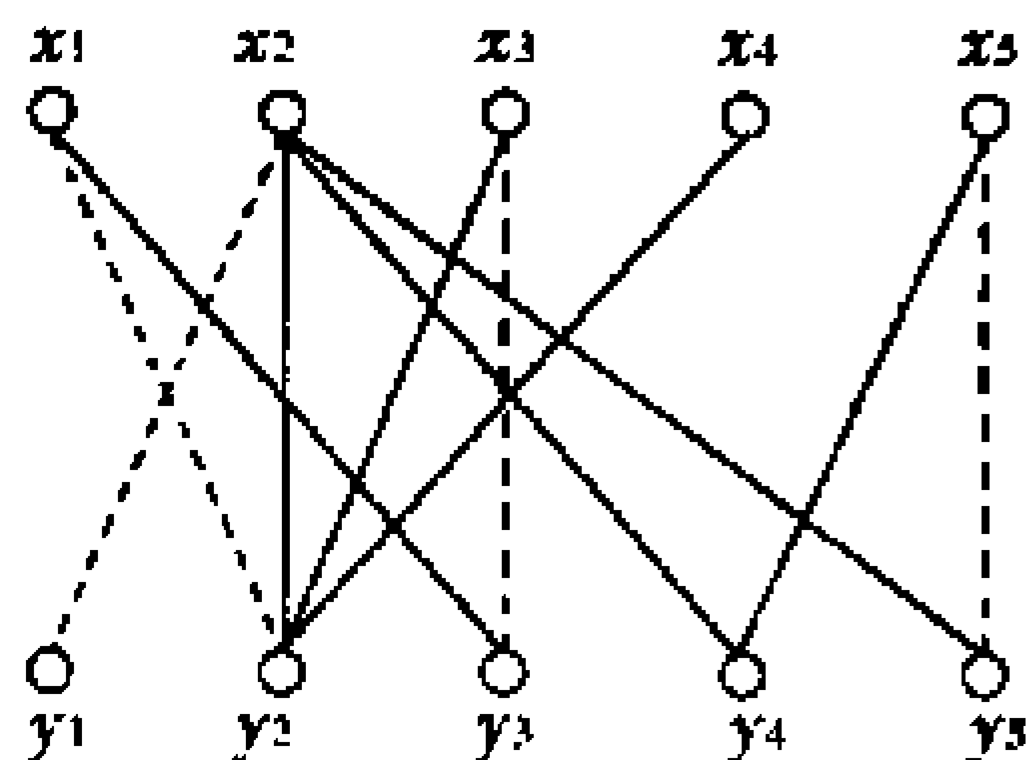


图 5-16

(3)$x=x_4$,得 $s=\{x_4,x_1,x_3\}$,$T=\{y_2,y_3\}$,$N_{G_l}(S)=T$,于是 $\alpha_l=\min\{l(x)+l(y)-w(x,y)\mid x\in S,y\in V_2-T\}=1$.

因而 x_1,x_2,x_3,x_4,x_5 的顶标分别为 4,2,3,0,3;y_1,y_2,y_3,y_4,y_5 的顶标分别为 0,1,1,0,0.

(4)用修改后的顶标 l' 得 $G_{l'}$ 及 $G_{l'}$ 的一个匹配(虚线),如图 5-17 所示. 此匹配即 $K_{5,5}$ 的最优匹配,其总权为 $2+4+1+4+3=14$.

当然,图 G 的最优匹配未必唯一. 例如,在上例中,完美匹配 $M'=\{x_1y_4,x_2y_1,x_3y_3,x_4y_2,x_5y_5\}$ 的权也为 14,显然 M' 也是上例中 $K_{5,5}$ 的最优匹配.

Kuhn-Munkres 算法可以用来求加权完全二分图 $(K_{n,n},w)$

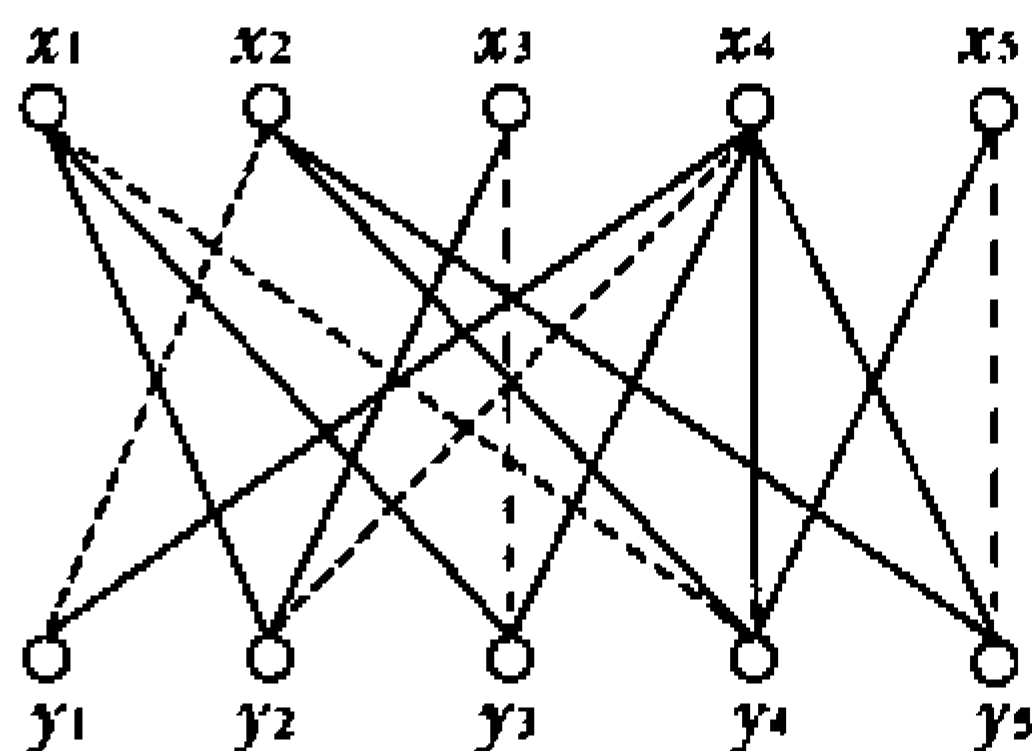


图 5-17

中总权最小的完美匹配.它是基于下列定理.

定理 2 设 a 是$(K_{n,n},w)$的加权矩阵 $A=(a_{ij})$中元素最大值,J_n 是 n 阶全 1 方阵,$A^*=(a_{ij}^*)=aJ_n-A$ 是$(K_{n,n},w^*)$的加权矩阵.则 M^* 是$(K_{n,n},w^*)$的权最大的完美匹配$\Leftrightarrow M^*$ 是$(K_{n,n},w)$的权最小的完美匹配,且 $w(M^*)=na-w^*(M^*)$.

证明 设 $K_{n,n}$是具有二划分(V_1,V_2)的完全二分图,其中 $V_1=\{x_1,x_2,\cdots,x_n\}$,$V_2=\{y_1,y_2,\cdots,y_n\}$,则 $K_{n,n}$有完美匹配,不妨设 $M=\{x_{i_1}y_{j_1},x_{i_2}y_{j_2},\cdots,x_{i_n}y_{j_n}\}$是 $K_{n,n}$的任一完美匹配,则

$$w^*(M)=\sum_{t=1}^{n}w^*(i_t,j_t)=\sum_{t=1}^{n}(a-w(i_t,j_t))$$
$$=na-\sum_{t=1}^{n}w(i_t,j_t)=na-w(M)$$

设 M^* 是$(K_{n,n},w^*)$的权最大的完美匹配,M'为$(K_{n,n},w)$的权最小的完美匹配,则

$$w^*(M')\leqslant w^*(M^*)=na-w(M^*)\leqslant na-w(M')=w^*(M')$$

因此,有 $w^*(M^*)=na-w(M')$,因而有

$$w^*(M')=w^*(M^*),w(M^*)=w(M')$$

并且 $w(M')=na-w^*(M')$,$w(M^*)=na-w^*(M^*)$.

例 2 求上例中所示加权图$(K_{5,5},w)$的权最小的完美匹配.

解 此时 $a=5$

$$A^{*}=aJ_{n}-A=\begin{bmatrix}2&0&0&1&4\\3&3&5&3&3\\3&1&1&4&5\\5&4&4&5&5\\4&3&4&2&2\end{bmatrix}$$

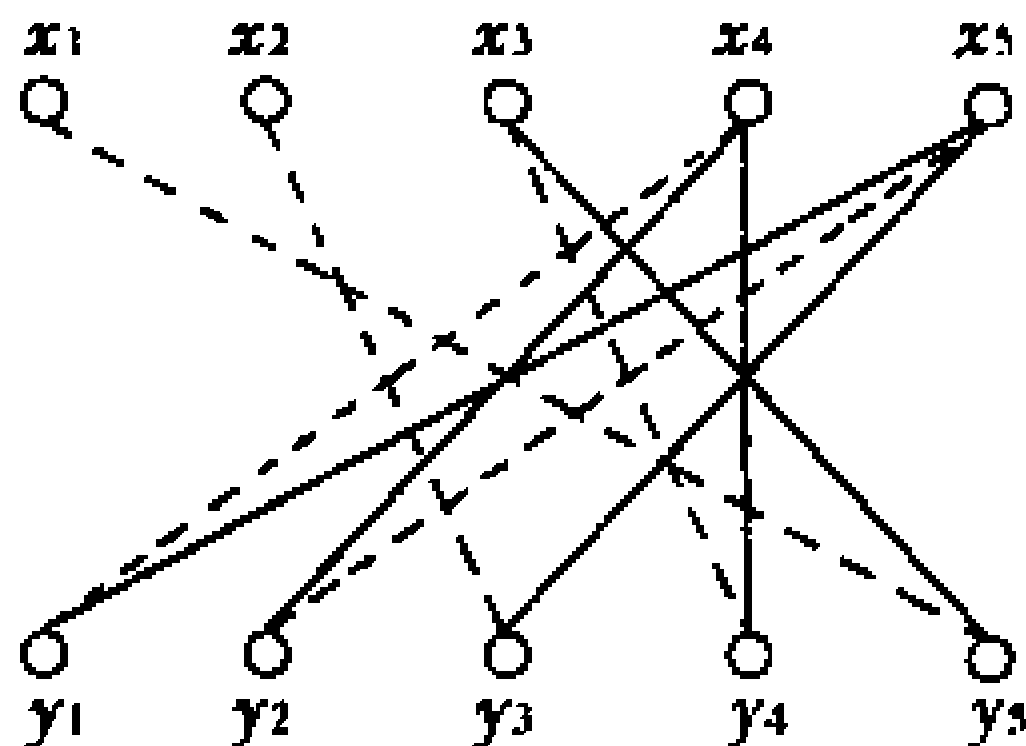


图 5-18

利用 Kuhn－Munkres 算法，从平凡标号开始，最后求得 $(K_{5,5},w^{*})$ 的最优匹配 M 如图 5-18 所示，虚线为匹配 M.

显然，此时 M 的权为 $4+5+4+5+3=21$. 再由定理 2 知，$(K_{5,5},w)$ 的权最小的完美匹配即为 $M=\{x_1y_5,x_2y_3,x_3y_4,x_4y_1,x_5y_2\}$，其权为 $5\times5-21=4$.

习 题 五

1. 设 G 是 n 阶任意图，α' 和 β' 分别为图 G 的匹配数和边覆盖数. 若 $\delta(G)>0$，则 $\alpha'+\beta'=n$.

2. 对任何二分图 G 有 $\alpha'(G)=\beta(G)$

3. 设 G 是二分图，α 和 β' 分别为图 G 的独立数和边覆盖数. 若 $\delta(G)>0$，则 $\alpha=\beta'$.

4. 称图 G 是 α 临界的（α－critical），如果对每条边 $e\in E(G)$ 均有 $\alpha(G-e)>\alpha(G)$. 证明

(1) α 临界图是简单图；

(2) 阶大于 2 的连通 α 临界图是 2 连通的.

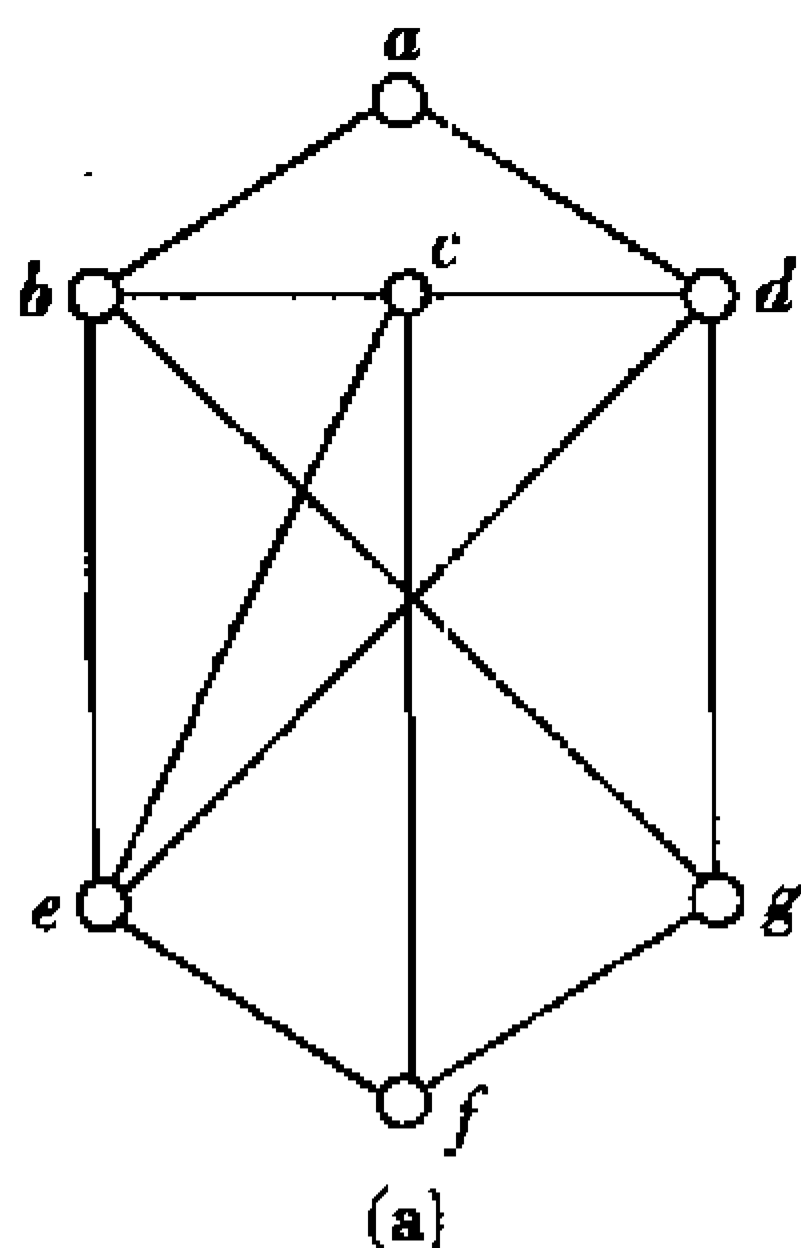
5. 称图 G 是 β 临界的(β-critical),如果对每条边 $e\in E(G)$ 均有 $\beta(G-e)<\beta(G)$.证明

(1) G 是 β 临界图 $\Leftrightarrow$ G 是 α 临界图;

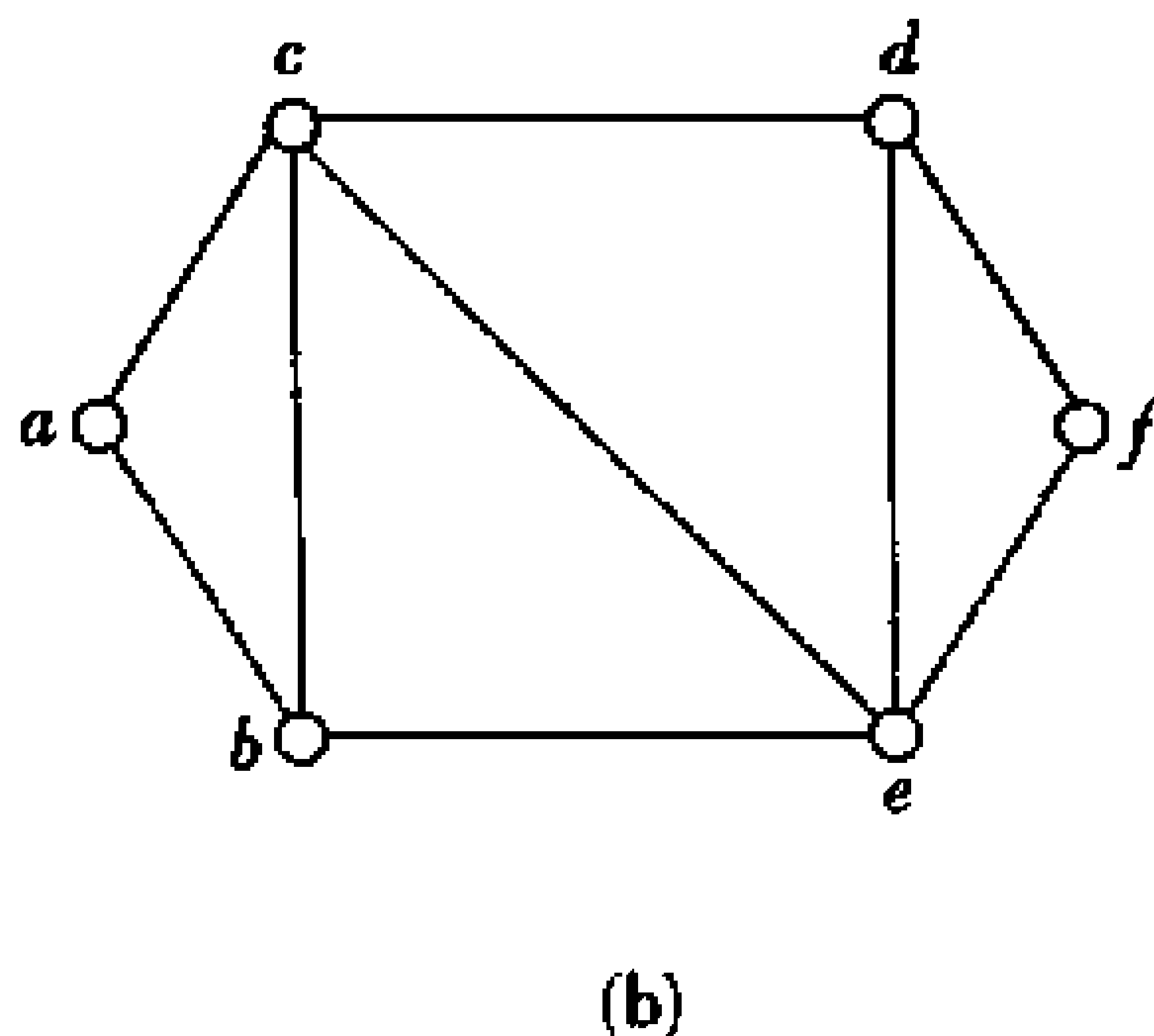
(2) 阶大于 2 的连通的 β 临界图是 2 连通的;

(3) 若 G 是连通的,则 $\beta\leqslant\frac{1}{2}(\varepsilon+1)$,其中 ε 为 G 的边数.

6. 求出图 5-19 所示两个图中的所有极小点覆盖和极大独立集,并写出最小点覆盖和最大独立集.



(a)



(b)

图 5-19

7. 求出图 5-19 所示中两个图的全部极小支配集.

8. 构造表示 $n\times n$ 棋盘的简单图,用边表示皇后对格子的控制,其中(1) $n=3$ (2) $n=4$.

9. 解释一下最小支配集的概念如何应用到确定控制 $n\times n$ 棋盘的最小皇后数的问题.

10. 求控制 $n\times n$ 棋盘的最小皇后数,其中

(1) $n=3$ (2) $n=4$ (3) $n=5$

11. 下列图的独立数是什么?

(1) K_n　　(2) C_n　　(3) Q_n　　(4) $K_{m,n}$

12. 证明:一个简单图里的顶点数小于或等于这个图的独立数与色数之积.

13. 证明:一个图的色数小于或等于 $v-\alpha+1$,其中 v 是这个图的顶点数,而 α 是这个图的独立数.

14. 证明:树至多有一个完美匹配.

15. 求 K_{2n} 和 $K_{n,n}$ 中不同完美匹配的个数.

16. 求图 5-20 所示图 G 的一个最大匹配.

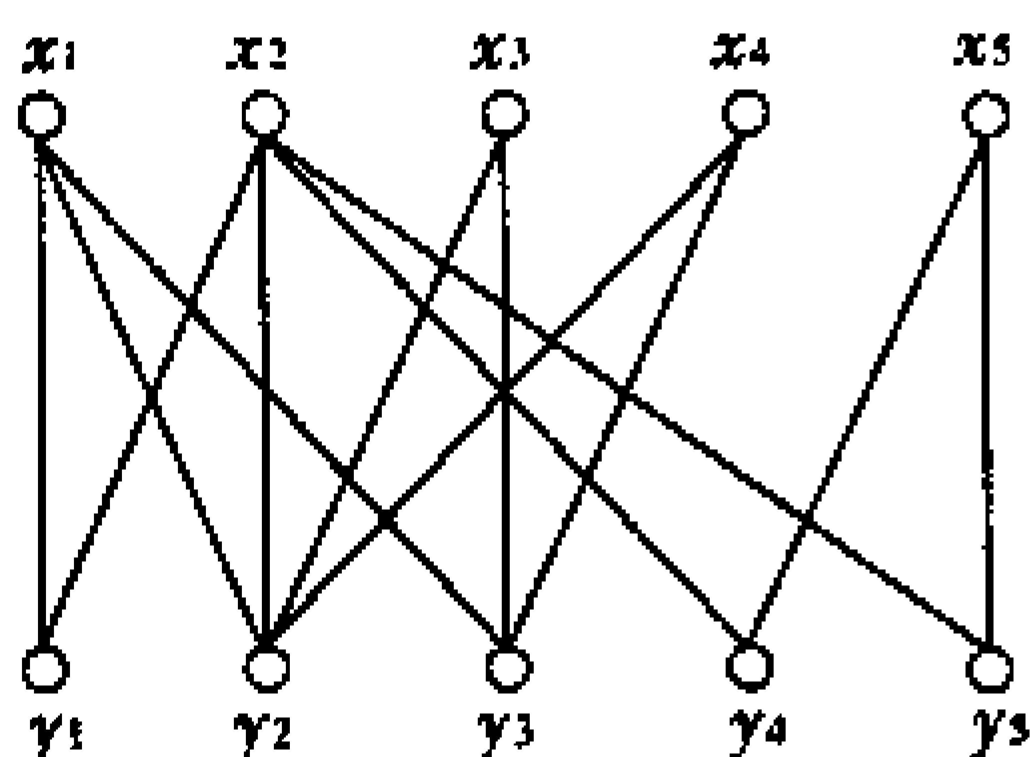


图 5-20

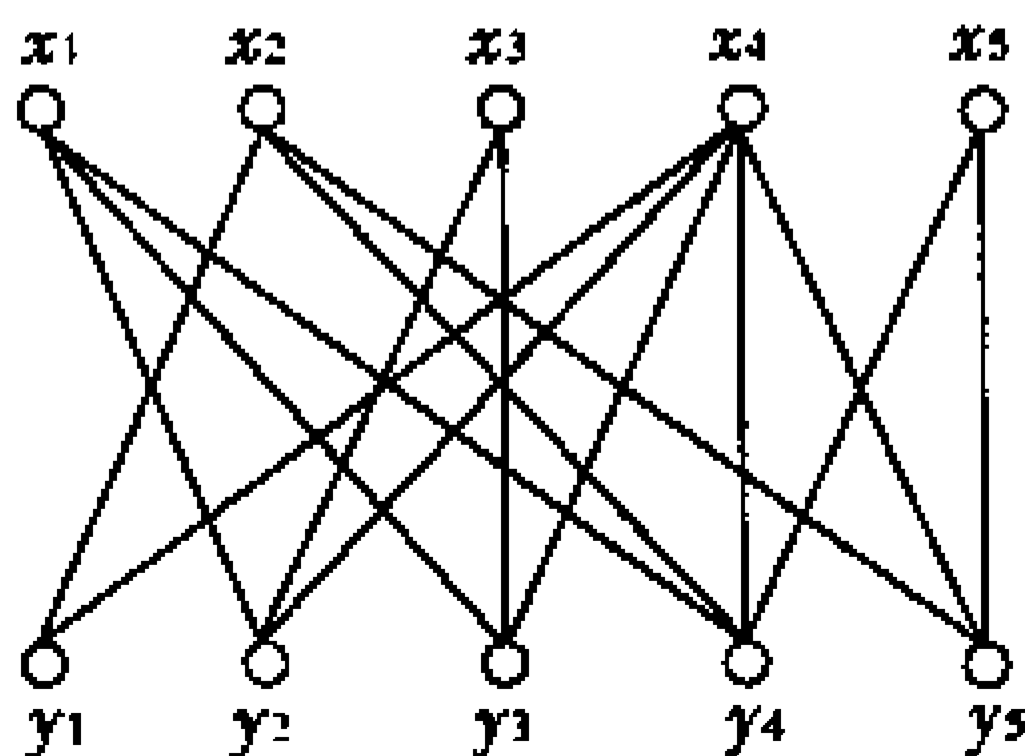


图 5-21

17. 求图 5-21 所示图 G 的一个完美匹配.

18. **工作排序问题**　有一台机床加工 n 种不同的零部件 J_i ($i=1,2,\cdots,n$). 每加工完一个零部件后,须将机床加以调整才能加工另一个零部件. 设加工完 J_i 后,在加工 J_j 之前机床调整时间为 t_{ij}. 问如何安排这些零部件的加工顺序使调整机床所耗总时间最短?

现有 6 种零部件需要在一台机床上加工. 设调整机床所消耗时间 t_{ij} 如下列矩阵表示:

$$T=(t_{ij})=\begin{bmatrix}0&5&3&4&2&1\\1&0&1&2&3&2\\2&5&0&1&2&3\\1&4&4&0&1&2\\1&3&4&5&0&5\\4&4&2&3&1&0\end{bmatrix}$$

求一个加工顺序使调整机床所消耗的总时间尽可能地短.

19. 求加权图$(K_{5,5},w)$和$(K_{5,5},w')$中权最大的与权最小的完美匹配.其中

$$w=\begin{bmatrix}9&8&5&3&2\\6&7&8&6&9\\5&8&1&4&7\\7&7&0&3&6\\9&8&6&4&5\end{bmatrix}\qquad w'=\begin{bmatrix}3&2&1&2&3\\1&4&2&1&2\\5&1&2&3&1\\3&2&6&4&1\\1&2&3&1&2\end{bmatrix}$$

20. 已知工人 x_1,x_2,x_3,x_4,x_5 做工作 y_1,y_2,y_3,y_4,y_5 的效率 w_{ij} 为下列矩阵所示

$$w=(w_{ij})=\begin{bmatrix}3&4&1&2&2\\0&1&2&3&1\\0&5&5&1&3\\2&2&1&0&5\\1&4&2&1&3\end{bmatrix}$$

(1)给出一种工作效率最大的分配方案.最大工作效率是多少?

(2)给出一种工作效率最低的分配方案.最低工作效率是多少?

21. 生成极大独立集的平移变换算法

(1)生存邻接矩阵;

(2)设一终止算法条件数 num=1;

(3)查找含 0 最多的某一行 k,将 k 与 1 进行一次平移变换并保存 $h[1]=k$;同时将此行中为 1 的元素结点号(即与 k 相邻)保存于 $h[2]\sim h[r]$中($r\leqslant n$);

(4)通过本章第一节定理 2 所述的变换过程找出所有含 $h[\mathrm{num}]$的极大独立集;

(5)num = num + 1,当 num$\leqslant r$ 时转到(4),否则停止.

编程实现上述算法,并通过实例上机检验运行效果.

22. 某人给 6 个人写了 6 封信,并且准备了 6 个写有收信地址的信封.问有多少种投放信笺的方法,使每份信笺与信封上收信人不相符?

23. 从 64 格棋盘上选出 16 格,使每行每列含有其中的 2 格,证明:可以把 16 个棋子(8 个白的与 8 个黑的)放置在所选的方格上,使每行每列都恰好有一个白的与一个黑的棋子.

24. 设 $G=\langle V_1,V_2,E\rangle$是二分图,证明:若对任何 $x\in V_1$,$y\in V_2$ 均有 $d_G(x)\geqslant d_G(y)$,则 G 有饱和 V_1 中所有顶点的匹配.

25. 设 G 至少包含 $2k$ 个顶点,其中每个顶点的度数至少是 k,则图 G 的最大匹配至少有 k 条边.

第六章　网络流和 Petri 网

现代社会可以说在很大程度上是通过各种网络来管理与控制的.网络可以是运输货物的运输网,输送石油的输油管道网,传送数据的计算机网,等等.网络除了有广泛的实际应用外,在理论上也有重大意义.有些较难处理的理论问题,用网络来研究将十分简洁清晰.

第一节　网络模型

定义 1　一个网络(network)是满足下列条件的简单加权有向图:

(1)一个入度为 0 的确定顶点,称为源(source);

(2)一个出度为 0 的确定顶点,称为汇(sink);

(3)有向边 $\langle i,j\rangle$ 的权 C_{ij} 是一个非负数,称为边 $\langle i,j\rangle$ 的容量(capacity).

例如,图 6－1 所示的有向图 G,该图表示一个输油管道网.原油在码头 a 卸下并通过网输送到炼油厂 Z.顶点 b、c、d 和 e 表示中间泵站.有向边表示系统的子管道并指出原油流动的方向.边上的标号指出子管道的容量.图 6－1 所示的有向图 G 就是一个网络,其中源是顶点 a 而汇是顶点 Z.边 $\langle a,b\rangle$ 的容量 C_{ab} 是 3,而边 $\langle b,c\rangle$ 的容量 C_{ac} 是 2.

定义 2　设 G 是一个网络.令 C_{ij} 表示有向边 $\langle i,j\rangle$ 的容量.G 的一个流(flow) F 赋与每个有向边 $\langle i,j\rangle$ 一个非负数 F_{ij},使得:

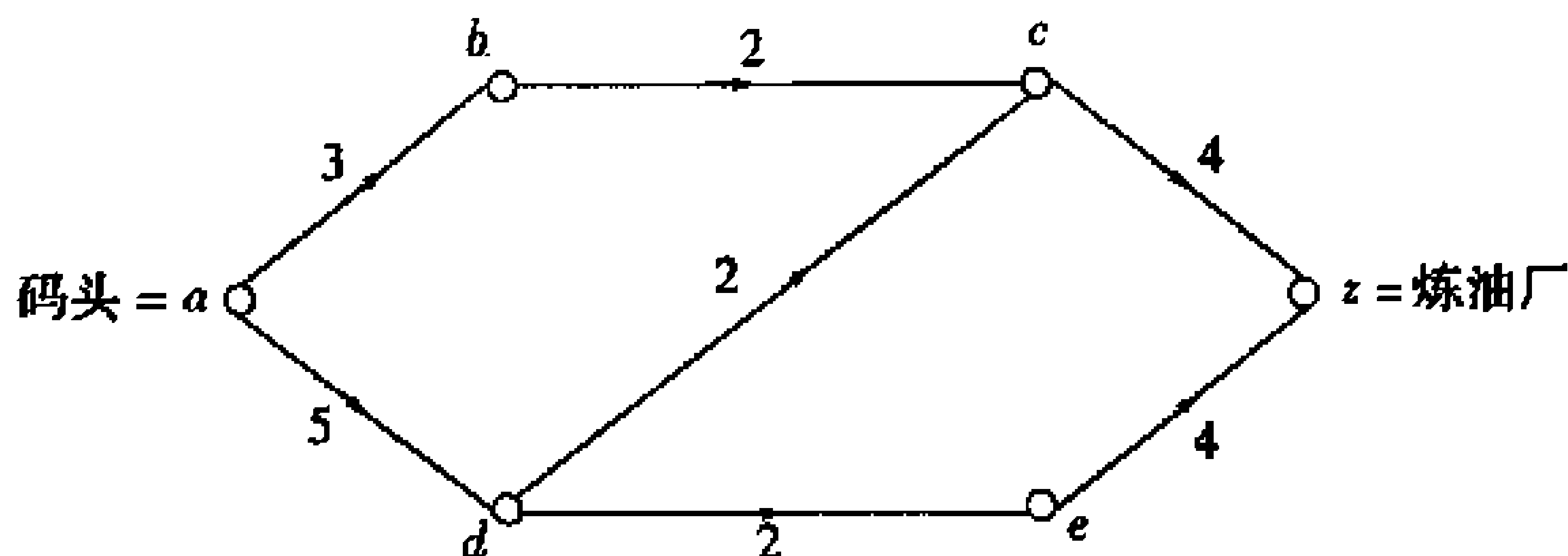


图 6-1　G

(1) $F_{ij} \leqslant C_{ij}$;

(2) 对于既不是源也不是汇的每个顶点 j, $\sum_i F_{ij} = \sum_i F_{ji}$. 其中,除非另有说明,总是对所有顶点 i 求和. 而且,如果 $\langle i,j \rangle$ 不是边,则设 $F_{ij} = 0$.

注释 1　(a) 网络 G 的流 F,也即函数 $F: E(G) \to R^+ \cup \{0\}$,其中 R^+ 为正实数集.

(b) 称 F_{ij} 是边 $\langle i,j \rangle$ 的**流量**.

(c) 称 $\sum_i F_{ij}$ **是流入 j 的流量或 j 的流入量**;称 $\sum_i F_{ji}$ **是流出 j 的流量或 j 的流出量**.

(d) 定义 2 中的条件(1)要求边 $\langle i,j \rangle$ 的流量不能超过该边的容量,所以称条件(1)为**相容条件**;定义 2 中的条件(2)说明任意中间点(即非源又非汇的顶点) j: j 的流入量等于 j 的流出量,所以称条件(2)为**守恒条件**.

(e) 若 $F_{ij} = C_{ij}$,即边 $\langle i,j \rangle$ 的流量已经达到它的容量,则称边 $\langle i,j \rangle$ 在流 F 下是**饱和的**,否则称边 $\langle i,j \rangle$ 是**不饱和的**.

定理 1　给定网络 G 的一个流 F,其源为 a,汇为 z,则流出源 a 的流量等于流入汇的流量,即

$$\sum_i F_{ai} = \sum_i F_{iz}$$

证明 设网络 G 的顶点集为 V，边集为 E，我们有

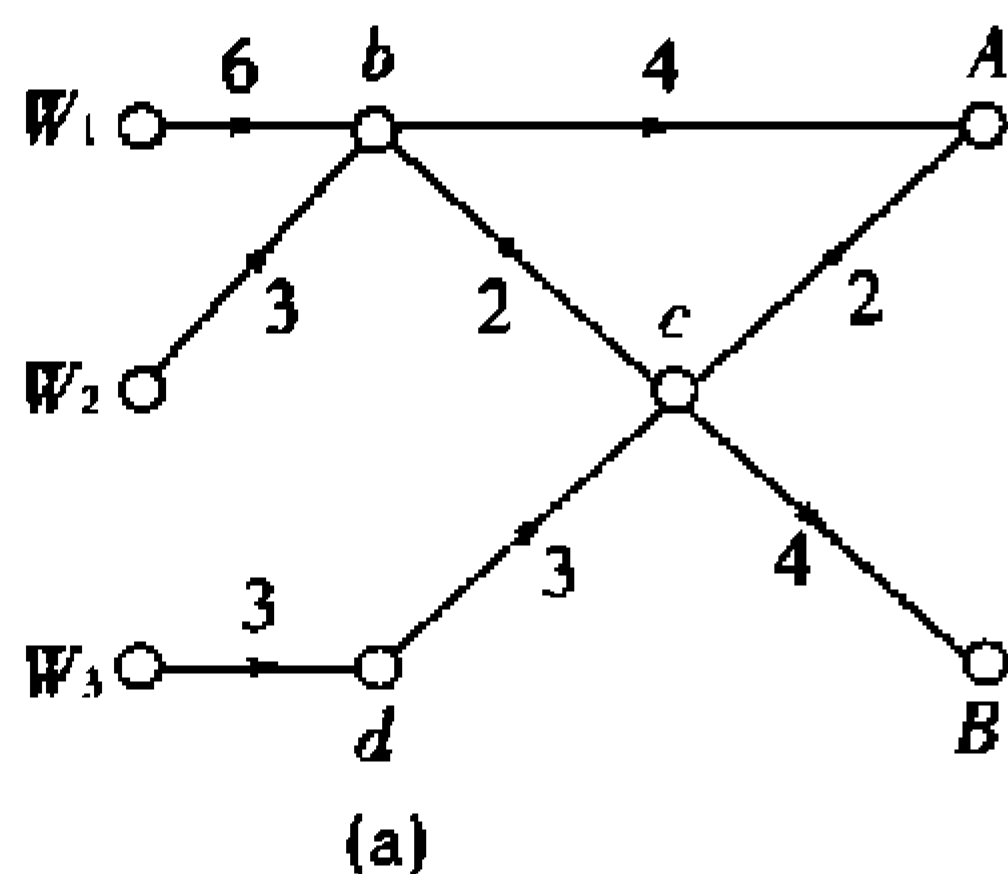
$$\sum_{j\in V}\sum_{i\in V}F_{ij}=\sum_{j\in V}\sum_{i\in V}F_{ji}$$

而每个二次和是 $\sum_{e\in E}F_e$，所以

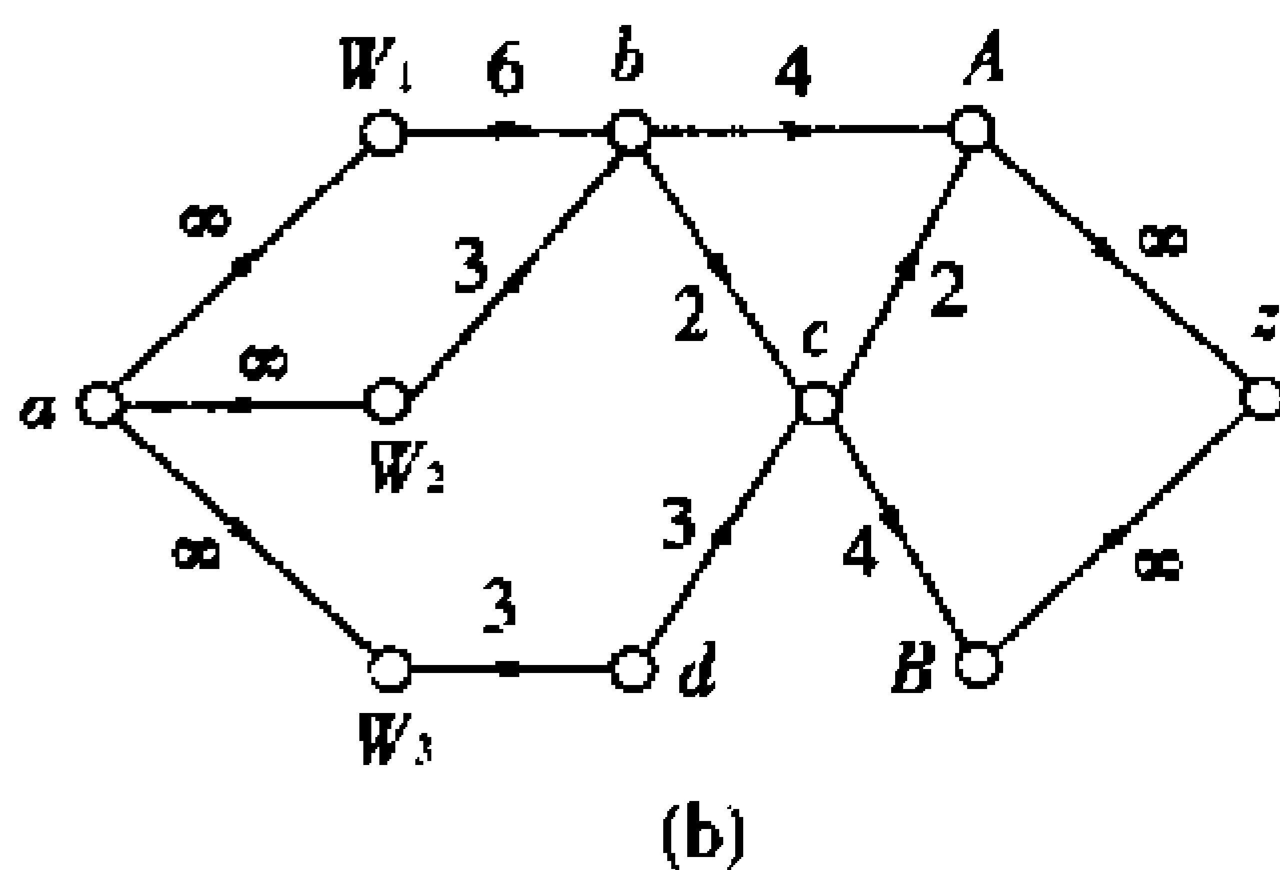
$$\begin{aligned}0&=\sum_{j\in V}\left(\sum_{i\in V}F_{ij}-\sum_{i\in V}F_{ji}\right)\\&=\left(\sum_{i\in V}F_{iz}-\sum_{i\in V}F_{zi}\right)+\left(\sum_{i\in V}F_{ia}-\sum_{i\in V}F_{ai}\right)\\&\quad+\sum_{\substack{j\in V\\ j\neq a,z}}\left(\sum_{i\in V}F_{ij}-\sum_{i\in V}F_{ji}\right)\\&=\sum_{i\in V}F_{iz}-\sum_{i\in V}F_{ai}\end{aligned}$$

由于对所有的 $i\in V$，$F_{zi}=0=F_{ia}$，且有如果 $j\in V-\{a,z\}$，则 $\sum_{i\in V}F_{ij}-\sum_{i\in V}F_{ji}=0$.

定义 3 设 F 是网络 G 的一个流，其源为 a，汇为 z，称值 $\sum_{i}F_{ai}=\sum_{i}F_{iz}$ 为流 F 的值，记作 $\mathrm{val}(F)$.



(a)



(b)

图 6-2

例 1 图 6-2(a)表示一个从三个井 W_1，W_2，W_3 为两个城市 A 和 B 供水的自来水网. 边上表明中间系统的容量. 顶点 b、c 和 d 代表中间泵站. 可将这个系统刻划为网络模型，为了得到一个确

定的源和汇,通过把几个源集结为一个**超级源**(**supersource**)和把几个汇集结为一个**超级汇**(**supersink**),我们可得到一个等价的网络.如图 6-2(b)所示.在图 6-2(b)中,∞代表一个无限容量.

例 2 在下午 6:00 到 7:00 期间,从 A 城可直接或通过 B 城到达 C 城,平均旅行时间是

A 到 B	15 分钟
B 到 C	30 分钟
A 到 C	30 分钟

道路上的最大容量是

A 到 B	3000 辆
B 到 C	2000 辆
A 到 C	4000 辆

将下午 6:00 到 7:00 从 A 城到 C 城的交通流表示为网络.

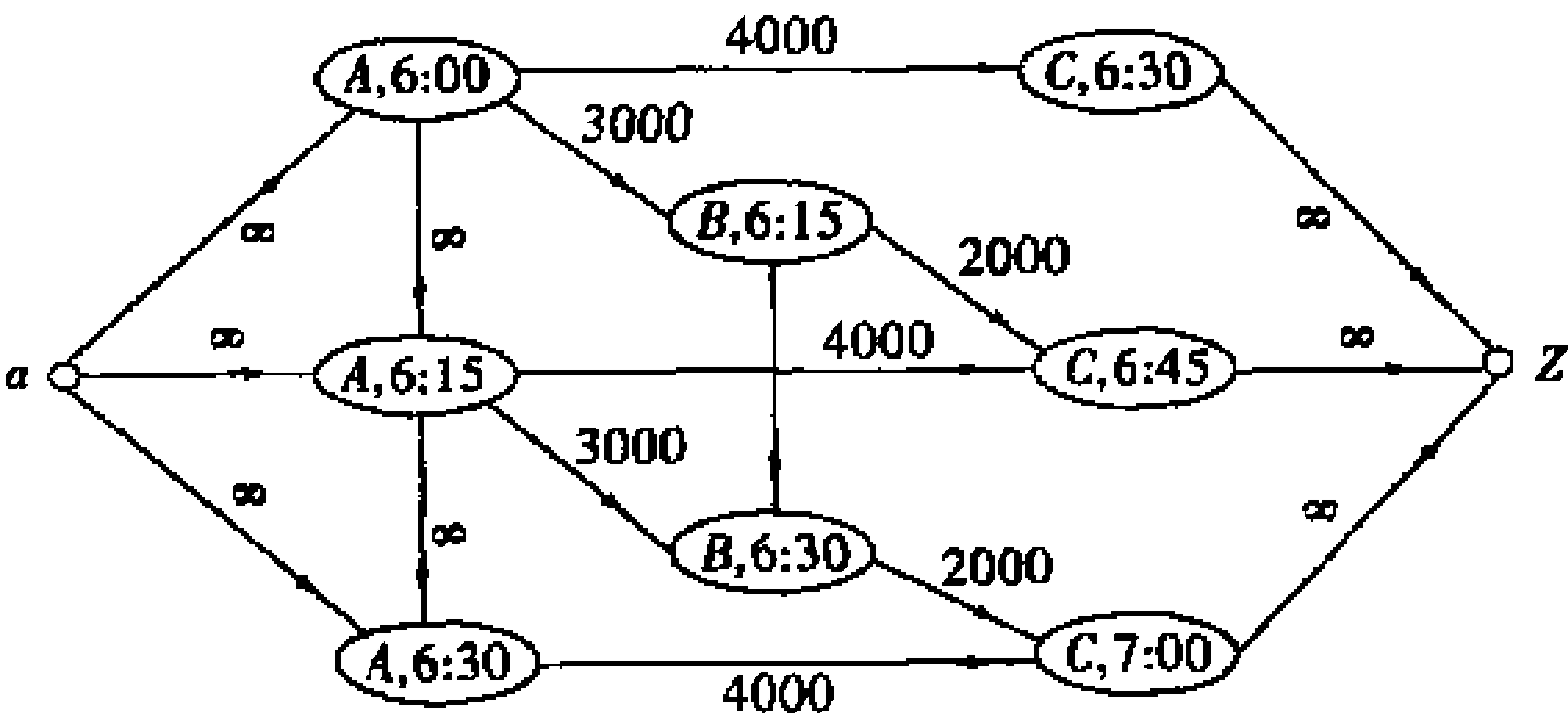


图 6-3

解 顶点代表特定时刻的城市.如果我们能够在下午的 t_1 离开 X 城且在下午的 t_2 到达 Y 城,则有一条边连结 X,t_1 和 Y,t_2.连接 A,t_1 和 A,t_2 及 B,t_1 和 B,t_2 的无限容量的边指出能够有任意辆汽车在 A 城或 B 城等待.最后,我们引入一个超级源和超级汇.如图 6-3 所示.

各种网络流问题已被用于有效的计算机网络的设计中．在描述计算机网络模型中，顶点是消息或交换中心，边表示数据在中心之间传输的通道，流是在通道上每秒传送的平均位数，边的容量是相应通道的容量．

如果 G 是一个网络，G 中的一个**最大流**(**maximal flow**)是具有最大值的流．一般地，可能存在几个具有相同最大值的流．在下一节，我们将给出一个求最大流的算法．下面我们再来介绍网络的切割．

定义 4　G 是一个网络，a 是源，z 是汇，$P\subseteq V(G)$，$\bar{P}=V(G)-P$，使得 $a\in P$，$Z\in\bar{P}$ 的$(P,\bar{P})$称为 G 的**切割**(**cut**)．

例如，图 6-4(a)的网络 G(边$\langle a,d\rangle$上的"3,2"依次为该边上的容量与流量)，如果我们设 $P=\{a,b,d\}$，则 $\bar{P}=\{c,e,f,z\}$，$(P,\bar{P})$是 G 的切割．如图，我们通过画一条划分顶点的线来表示一个切割．图 6-4(b)的网络 G，设 $P=\{a,b,d\}$，则 $\bar{P}=\{c,e,z\}$，$(P,\bar{P})$是网络 G 的一个切割．

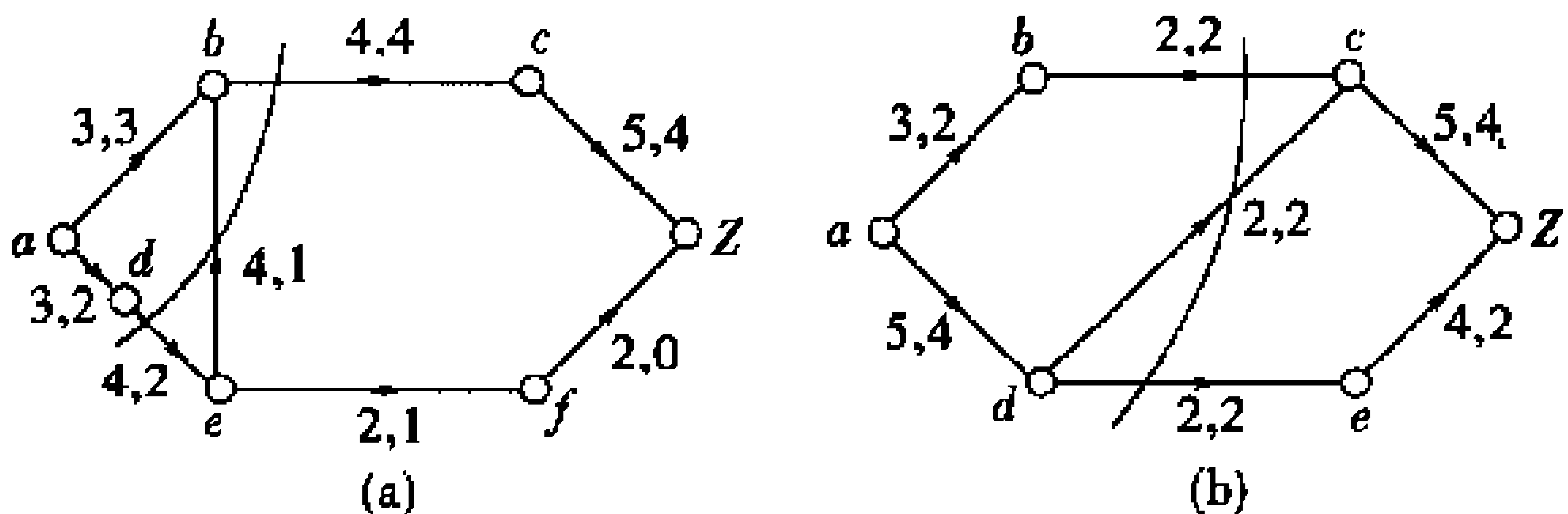


图 6-4

定义 5　网络 G 的切割$(P,\bar{P})$的容量是数值 $C(P,\bar{P})=\sum_{i\in P}\sum_{j\in\bar{P}}C_{ij}$，具有最小容量的切割称为**最小切割**．

例如，图 6-4(a)的切割的容量是 $C_{bc}+C_{de}=4+4=8$，而图 6-4(b)的切割的容量是 $C_{bc}+C_{dc}+C_{de}==2+2+2=6$．

定理 2 设 F 是网络 G 的流，$(P,\bar{P})$ 是网络 G 的切割，则 $(P,\bar{P})$ 的容量大于或等于 F 的值，即

$$\mathrm{val}(F)\leqslant C(P,\bar{P})$$

证明 我们知道 $\sum_{j\in P}\sum_{i\in P}F_{ji}=\sum_{j\in P}\sum_{i\in P}F_{ij}$

于是

$$\begin{aligned}\mathrm{val}(F)&=\sum_i F_{ai}=\sum_{j\in P}\sum_i F_{ji}-\sum_{j\in P}\sum_i F_{ij}\\&=\sum_{j\in P}\sum_{i\in P}F_{ji}+\sum_{j\in P}\sum_{i\in\bar{P}}F_{ji}-\sum_{j\in P}\sum_{i\in P}F_{ij}-\sum_{j\in P}\sum_{i\in\bar{P}}F_{ij}\\&=\sum_{j\in P}\sum_{i\in\bar{P}}F_{ji}-\sum_{j\in P}\sum_{i\in\bar{P}}F_{ij}\leqslant\sum_{j\in P}\sum_{i\in\bar{P}}F_{ji}\leqslant\sum_{j\in P}\sum_{i\in\bar{P}}C_{ji}\end{aligned}$$

定理 3 设 F 是网络 G 的一个流，$(P,\bar{P})$ 是网络 G 的一个切割.

(1)若 $C(P,\bar{P})=\mathrm{val}(F)$，则流 F 最大且切割 $(P,\bar{P})$ 最小.

(2) $C(P,\bar{P})=\mathrm{val}(F)$，当且仅当

$$\begin{cases}F_{ij}=C_{ij} & i\in P,j\in\bar{P}\\F_{ij}=0 & i\in\bar{P},j\in P\end{cases}$$

证明 (1)假设 F^* 是网络 G 的最大流，且 $(P^*,\overline{P^*})$ 是网络 G 的最小切割，则 $\mathrm{val}(F^*)\leqslant C(P^*,\overline{P^*})$. 由最大流和最小切割的定义知，$\mathrm{val}(F)\leqslant\mathrm{val}(F^*)$ 且 $C(P^*,\overline{P^*})\leqslant C(P,\bar{P})$ 从而

$$\mathrm{val}(F)\leqslant\mathrm{val}(F^*)\leqslant C(P^*,\overline{P^*})\leqslant C(P,\bar{P})$$

因为 $\mathrm{val}(F)=C(P,\bar{P})$，故 $\mathrm{val}(F)=\mathrm{val}(F^*)$，$C(P^*,\overline{P^*})=C(P,\bar{P})$，所以 F 是最大流且 $(P,\bar{P})$ 是最小切割.

(2)由本节定理 2 的证明知，当 $\sum_{j\in P}\sum_{i\in\bar{P}}F_{ij}=0$ 和 $\sum_{j\in P}\sum_{i\in\bar{P}}F_{ji}=\sum_{j\in P}\sum_{i\in\bar{P}}C_{ji}$ 时，有 $C(P,\bar{P})=\mathrm{val}(F)$，于是结论得证.

在图 6-4(b)中，流的值和切割容量都是 6，所以，流量最大且切割最小.

如图 6-5 所示的源为 a 汇为 z 的网络 G,流 F 的值为 4. 如果 $P=\{a,b,c\}$,则 $(P,\bar{P})$ 是 G 的切割,且 $C(P,\bar{P})=4$,因此 F 是 G 的最大流且 $(P,\bar{P})$ 是 G 的最小切割.

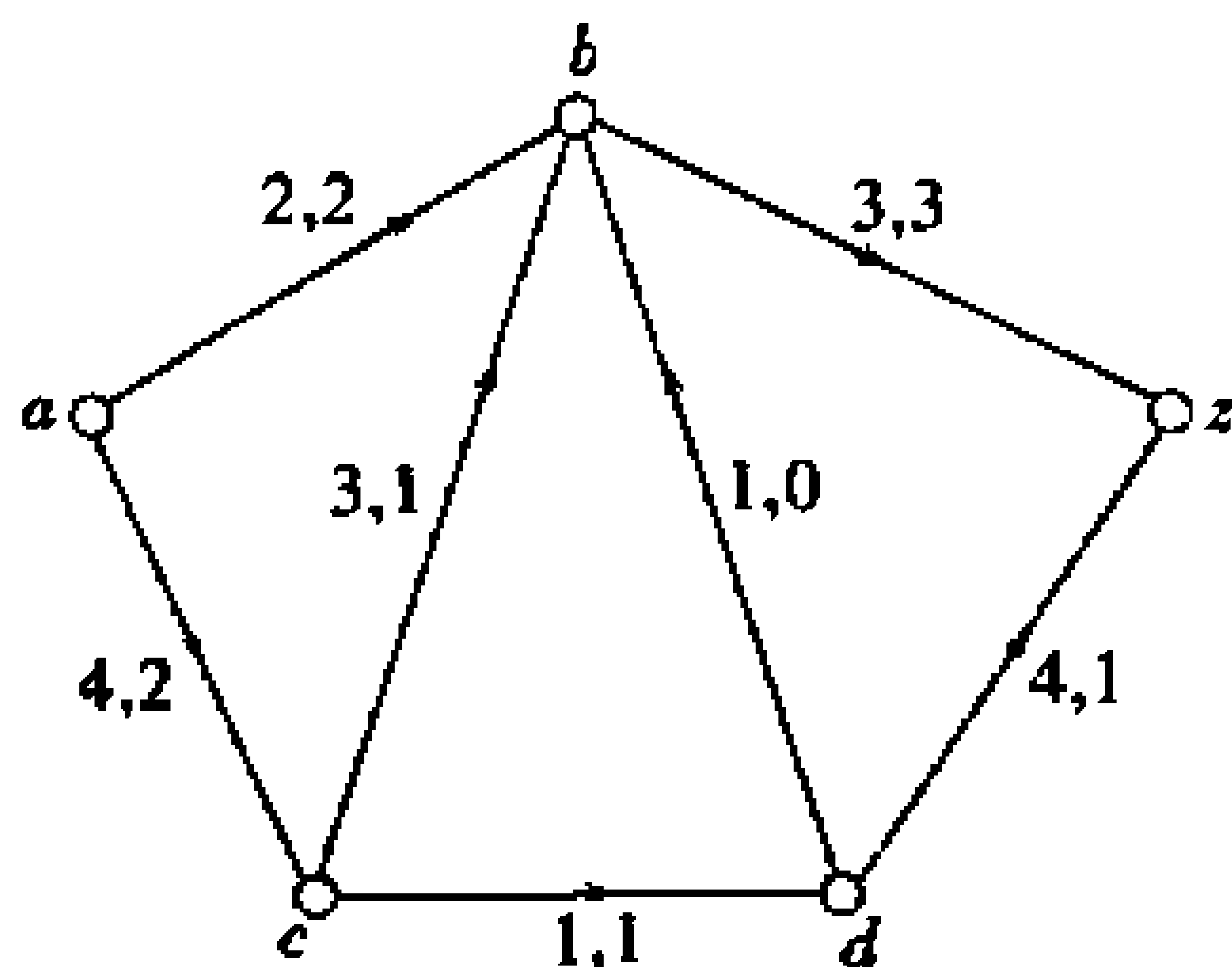


图 6-5

第二节　最大流算法

对于网络 G 问题可描述为:在 G 中求出一个最大流. 本节将给出求得最大流的一个算法,其基本概念较简单. 即从某个初始流开始,重复地增加流的值到不能再改进为止,则最后流将是一个最大流. 为此,不妨将每条边上的流量设置为 0 作为初始流量. 为了增加给定流量的值,我们必须找出从源到汇的一条路并沿这条路增加流量.

G 是具有源 a、汇 z 和容量 C 的一个网络,$P=v_0e_1v_1e_2v_2\cdots v_{n-1}e_nv_n$, $v_0=a$, $v_n=z$ 是 G 中从 a 到 z 的一条路. 规定路 P 的方向由源 a 指向汇 z,则 P 中与规定方向一致的边称为**正向边**,否则称为**反向边**.

如果我们能够找到一条路 P,使 P 上每条边都是正向边,并且每条边上流量小于其容量,就可能增加流量值.

例如,图 6-6(a)所示为从源 a 到汇 z 的一条路 P,路 P 上每条边都是正向的.在这个网络中流的值能够增加 1,如图 6-6(b)所示.

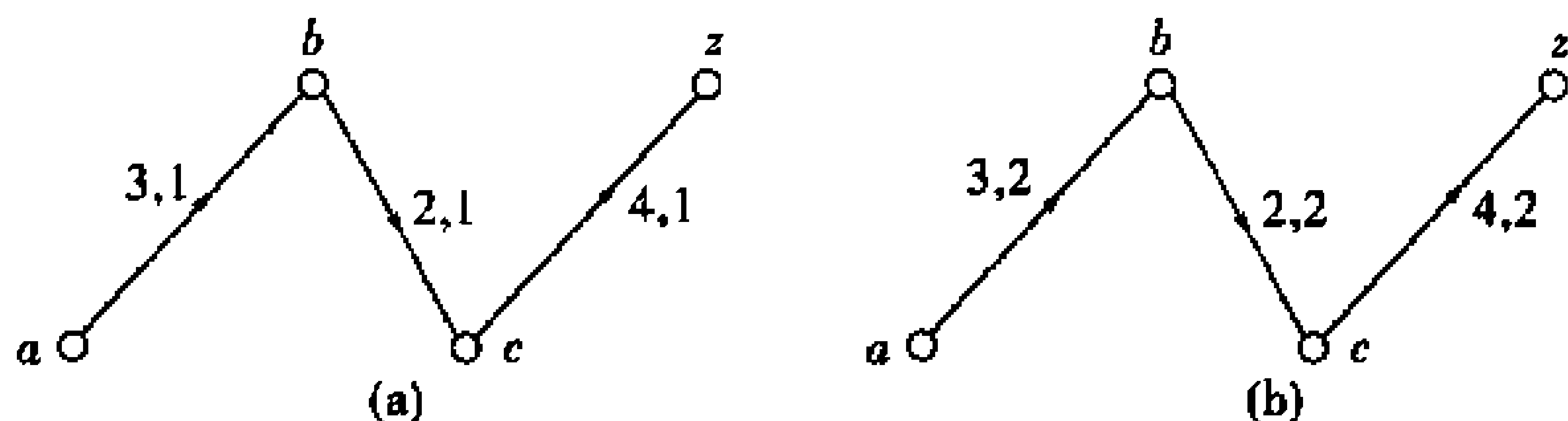


图 6-6

在某些含有正向边和反向边的从源到汇的路上也有可能增加流的值.设 P 是从源 a 到汇 z 的一条路,并且 x 是 P 中的一个顶点,且 x 既不是 a,也不是 z.

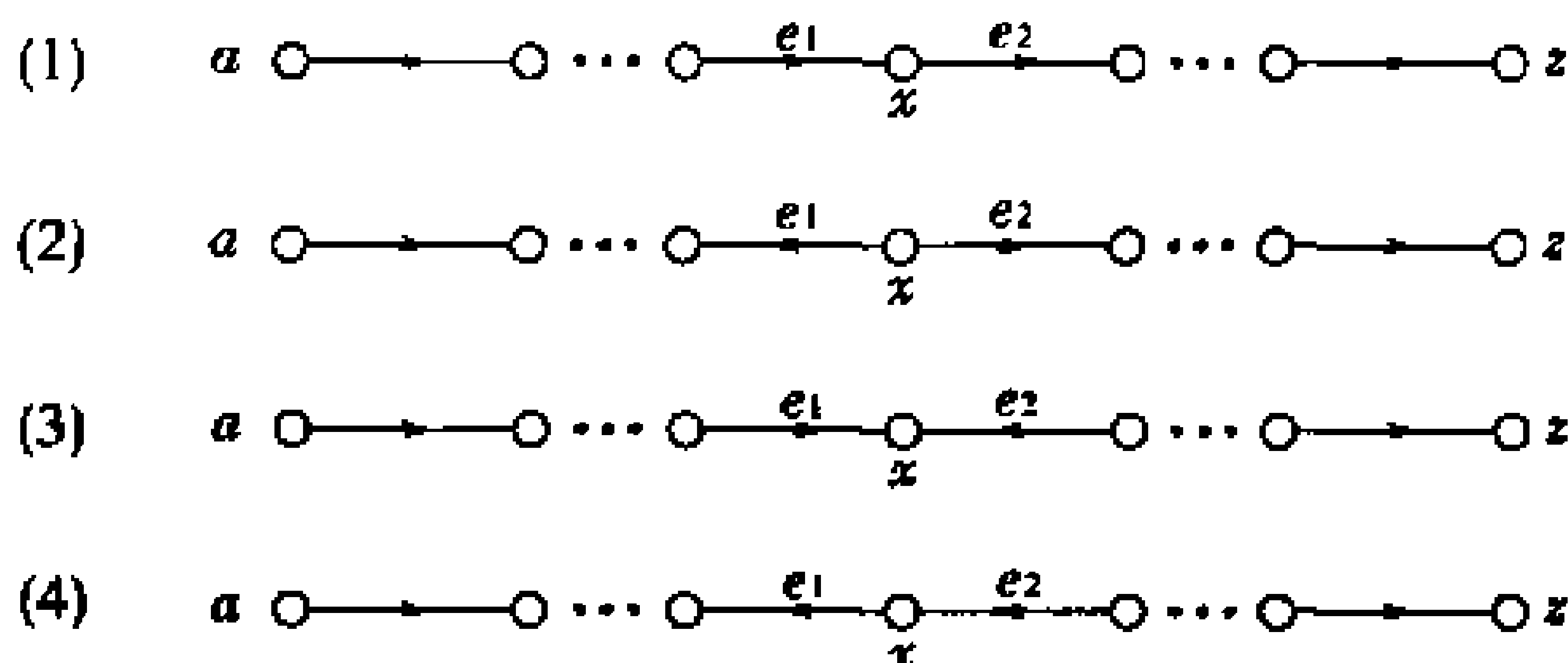


图 6-7

关联 x 的边 e_1 和 e_2 有四种情形,如图 6-7 所示.在情形(1),两条边都是正向的.此时,在每条边上增加流量 Δ,进入 x 的流量值始终等于离开 x 的流量值.在情形(2),如果我们在 e_2 增加流量 Δ,则必须在 e_1 减少流量 Δ,才能使进入 x 的流量值等于离开 x 的流量值.情形(3),只要在 e_1 增加流量 Δ,在 e_2 减少流量 Δ.情形(4),在两边 e_1 和 e_2 都减少流量 Δ.在每种情形,结果的边赋值都给出一个流量.当然,为了实现这些变更,我们必须在正向边上

流量小于容量，而在反向边上有非负流量.

例如，图 6-8(a)所示为源 a 到汇 z 的一条路. 边 $\langle a,b\rangle$，$\langle c,d\rangle$ 和 $\langle d,z\rangle$ 为正向边，而 $\langle c,b\rangle$ 为反向边. 我们在反向边 $\langle c,b\rangle$ 上减少流量 1，在正向边 $\langle a,b\rangle$，$\langle c,d\rangle$ 和 $\langle d,z\rangle$ 增加流量 1，如图 6-8(b)所示. 新的流的值比原先的流的值大 1.

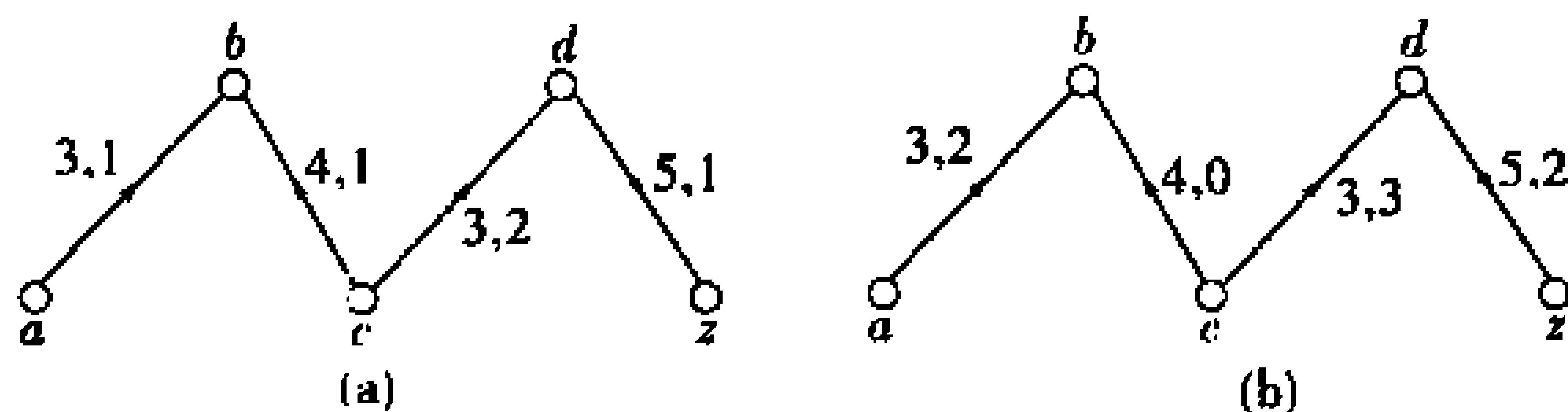


图 6-8

为此，我们得到下面定理.

定理 1 设 F 和 C 分别为网络 G 的流和容量，P 是网络 G 中从源 a 到汇 z 的满足下列条件的路：(1)对 P 中每条正向边 $\langle i,j\rangle$，$F_{ij}<C_{ij}$；(2)对 P 中每条反向边 $\langle i,j\rangle$，$F_{ij}>0$.

设 $\Delta=\min X$，这里 X 由数 $C_{ij}-F_{ij}$（对于 P 中的正向边 $\langle i,j\rangle$）和 F_{ij}（对于 P 中的反向边）组成. 定义

$$F_{ij}^{*}=\begin{cases}F_{ij} & \langle i,j\rangle\text{不在 }P\text{ 中}\\ F_{ij}+\Delta & \langle i,j\rangle\text{为正向边}\\ F_{ij}-\Delta & \langle i,j\rangle\text{为反向边}\end{cases}$$

则 F^{*} 是网络 G 的一个流，它的值比流 F 的值大 Δ.

证明 由显示于图 6-6、图 6-7 和图 6-8 的例题可知，F^{*} 是网络 G 的流. 因为 P 中的边 $\langle a,v\rangle$ 的流量增加 Δ，所以 F^{*} 的值比 F 的值大 Δ. 证毕

设 F 和 C 分别为网络 G 的流和容量，a 与 z 分别为网络 G 的源与汇. x 是 G 的顶点，且 $x\neq a$，并设 P 是 G 中一条连接 a 和 x 的路，给定 P 从 a 到 x 的方向为正向，若

(1)对 P 中每条正向边$\langle i,j\rangle$,$F_{ij}<C_{ij}$

(2)对 P 中每条负向边$\langle i,j\rangle$,$F_{ij}>0$.

则称 P 是非 F 饱和路(unsaturated path),否则,称 P 是**F 饱和路(saturated path)**. 非 F 饱和的 az 路 P 称为 **F 可增路(increment path)**.

定理 2 设网络 G 的源为 a 汇为 z,F 和 C 分别为 G 的流和容量,则 F 是最大流当且仅当 G 中不存在 F 可增路.

证明 ($\Rightarrow$)假设 G 中存在 F 可增路 P,则由定理 1 知,存在 G 的流 F^*,使得 F^* 的值比 F 的值大 Δ,而 $\Delta>0$,这与 F 是最大流矛盾.

($\Leftarrow$)设 $S\subseteq V(G)$,且对 S 中的每个顶点 x 存在一条非 F 饱和路 ux 路,显然,$u\in s$. 又已知 G 中不存在 F 可增路,故 $Z\notin S$,令 $k=(s,\bar{s})$是 G 的切割. 对任意边$\langle y,w\rangle\in k$,且 $y\in s$,$w\in\bar{s}$,所以存在一条非 F 饱和路 uy,因此,$F_{yw}=C_{yw}$,否则向上述 uy 路中添加边$\langle y,w\rangle$将得非 F 饱和路 uw,这与 $w\notin S$ 矛盾. 类似地,对任意边$\langle y,w\rangle\in k$,且 $y\in\bar{s}$,$w\in s$,有 $F_{yw}=0$. 由上节定理 3 知,F 是最大流. 证毕

到此,我们可以构造求网络 G 的最大流的算法,其步骤为:

(1)从一个流(例如,每条边上的流量为 0)开始;

(2)查找 F 可增路. 如果 F 可增路不存在,结束,流是最大流;

(3)沿 F 可增路增加流量 Δ,其中 Δ 按定理 1 所定义,并转向(2).

算法 求网络的最大流

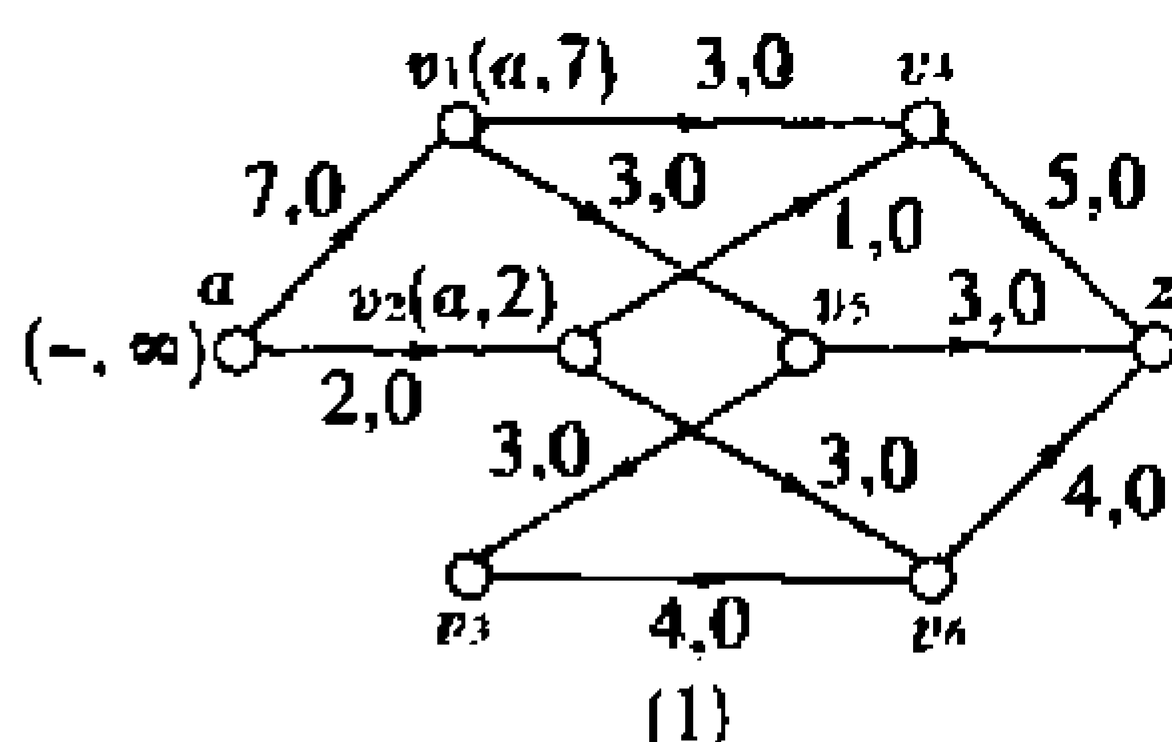
这个算法求出一个网络的最大流. 每条边的容量是一个非负数.

输入:一个网络,它具有源 a,汇 z,容量 C,顶点 $a=v_0,\cdots,v_n=Z$,和 n

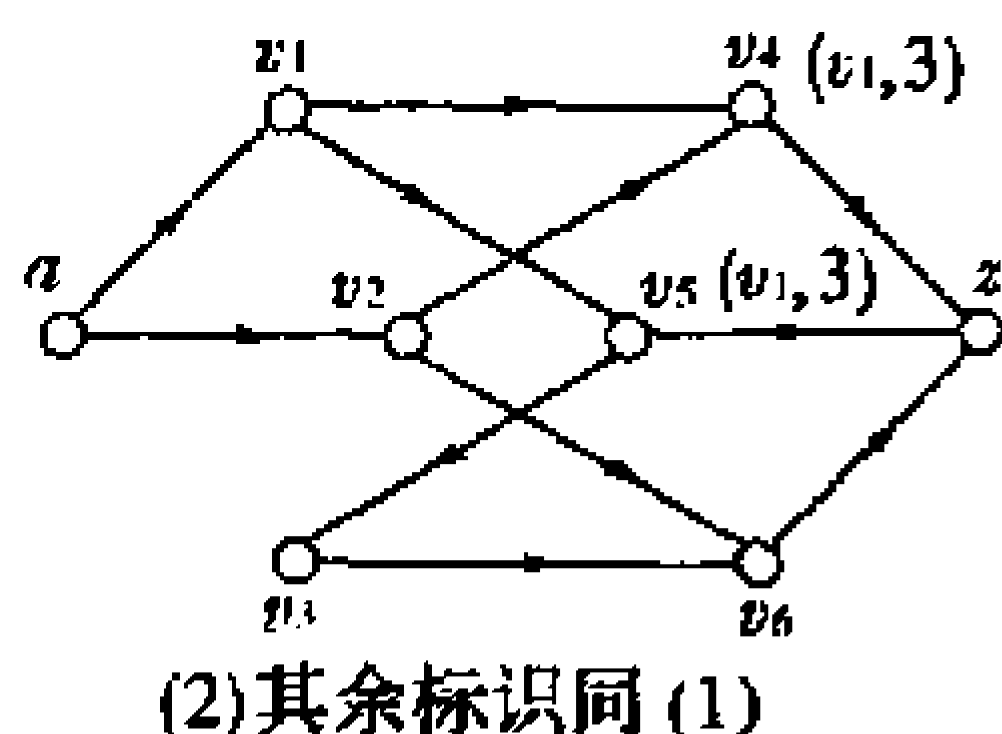
```
输出：最大流 F
procedure man-flow  (a, z, C, v, n)
{v 的标记是(P(v), val(v)},用 0 流量开始}
   for  每条边〈i, j〉  do
      F_ij := 0
   for  i := 0  to  n  do
   begin
      P(v_i) := null
      val(v_i) := null
   end
{标记 a}
   p(a) := -
   val(a) := ∞
{s 是未考查的有标记的顶点}
   s := {a}
{继续，直到 z 被标记}
   While  val(z) = null  do
      begin
         if  s = ∅  then  输出最大流 F
         在 s 中选择 v
         s := s - {v}
         Δ := val(v)
         for  val(w) = null  的每条边〈v, w〉  do
            if  F_vw < C_vw  then
               begin
                  p(w) := v
                  val(w) := min{Δ, C_vw - F_vw}
```

```
            s:=s∪{w}
          end
      for  val(w)=null  的每条边⟨w,v⟩  do
        if  F_wv>0  then
          begin
            p(w):=v
            val(w):=min{Δ,F_uv}
            s:=s∪{w}
          end
  end
{求出从 a 到 z 的路,修改其上的流量}
  W_0:=z
  k:=0
  While  W_k≠a  do
    begin
      W_{k+1}:=P(W_k)
      k:=k+1
    end
  P:=(W_k,W_{k-1},…,W_1,W_0)
  △:=val(z)
  for  i:=1  to  k  do
    begin
      e:=⟨W_i,W_{i-1}⟩
      if  e 在 P 中是正向边
        then  F_e:=F_e+△
        else  F_e:=F_e-△
      end
```

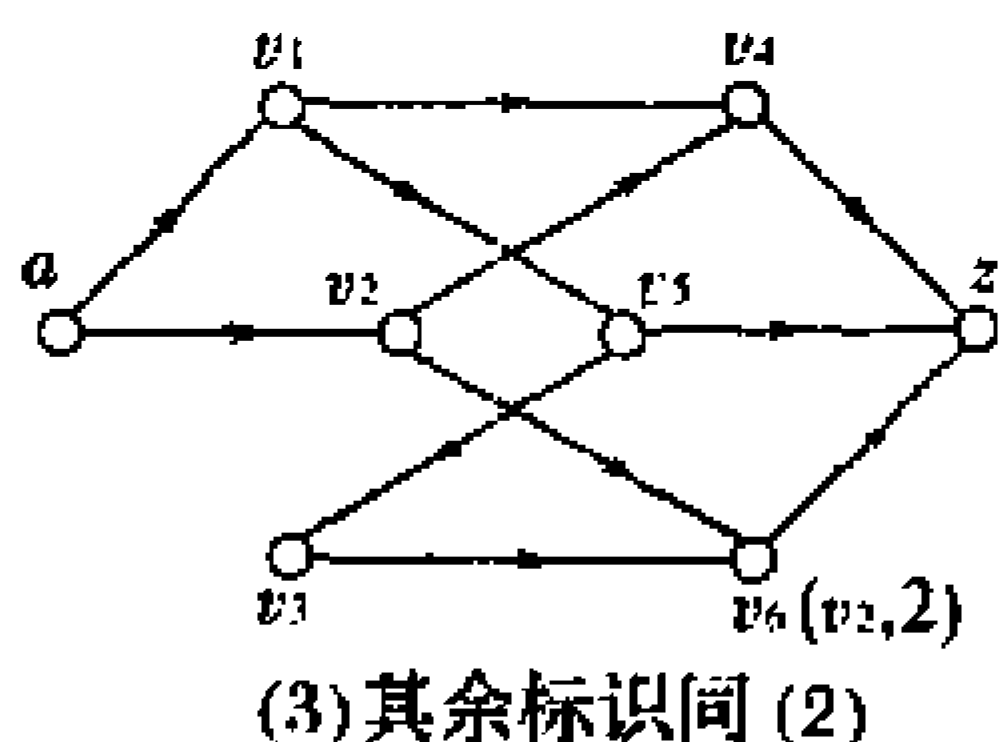
例 1　在这个讨论中，如果顶点 v 满足 $P(v)=q$ 和 $\mathrm{val}(v)=t$，则我们在图中标记 v 为 (q,t)。我们初始化每条边的流量为 0。其解题过程显示于图 6-9(1)～(16)。



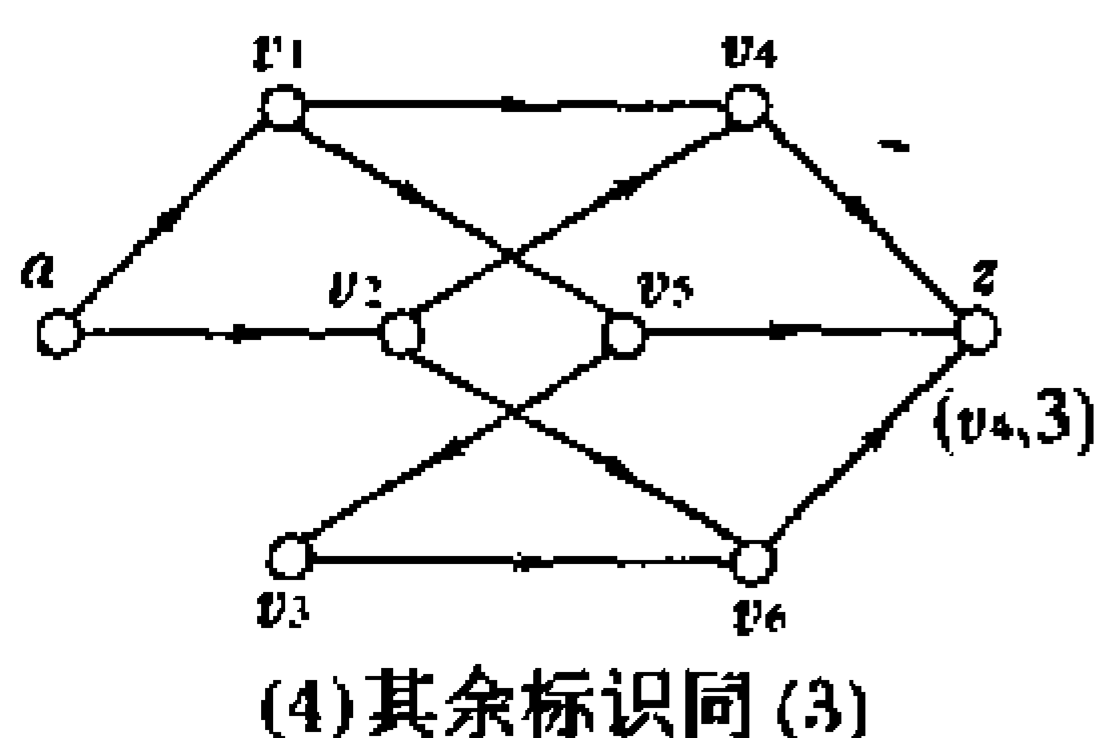
(1)



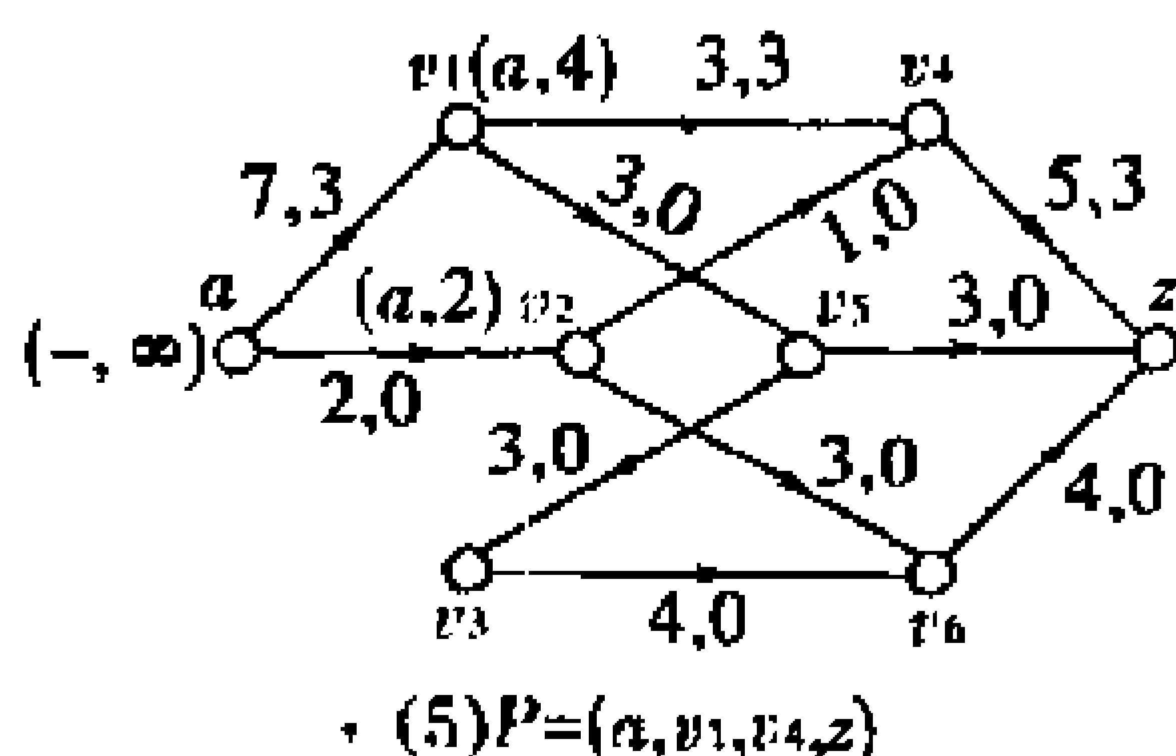
(2)其余标识同(1)



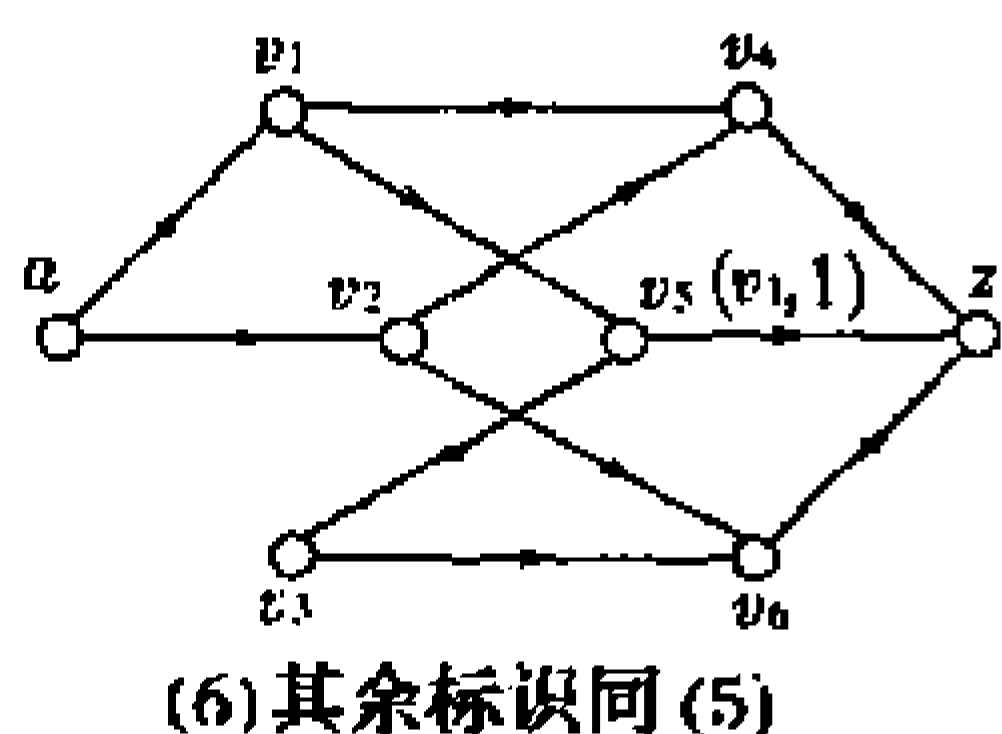
(3)其余标识同(2)



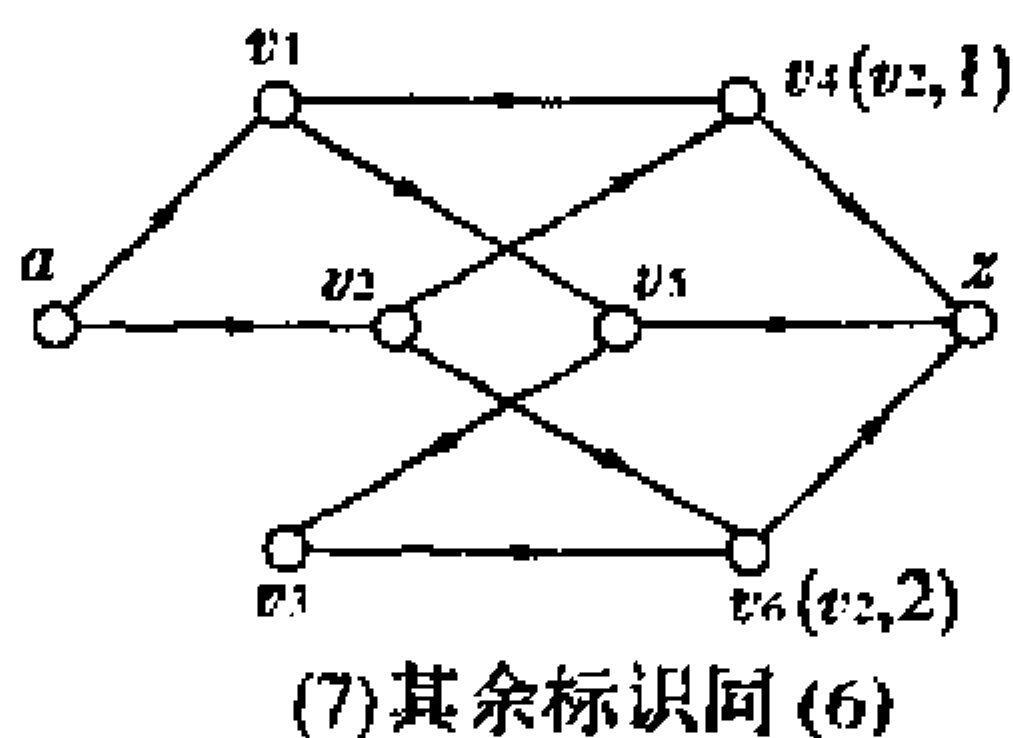
(4)其余标识同(3)



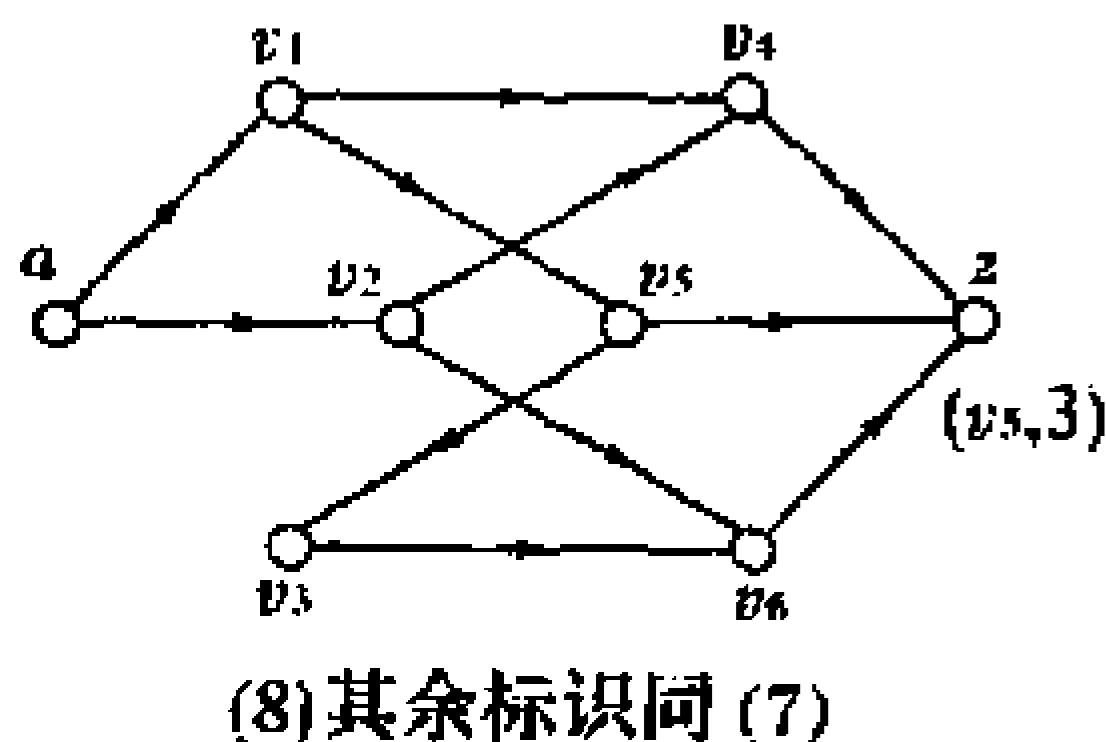
(5)$P=(a,v_1,v_4,z)$



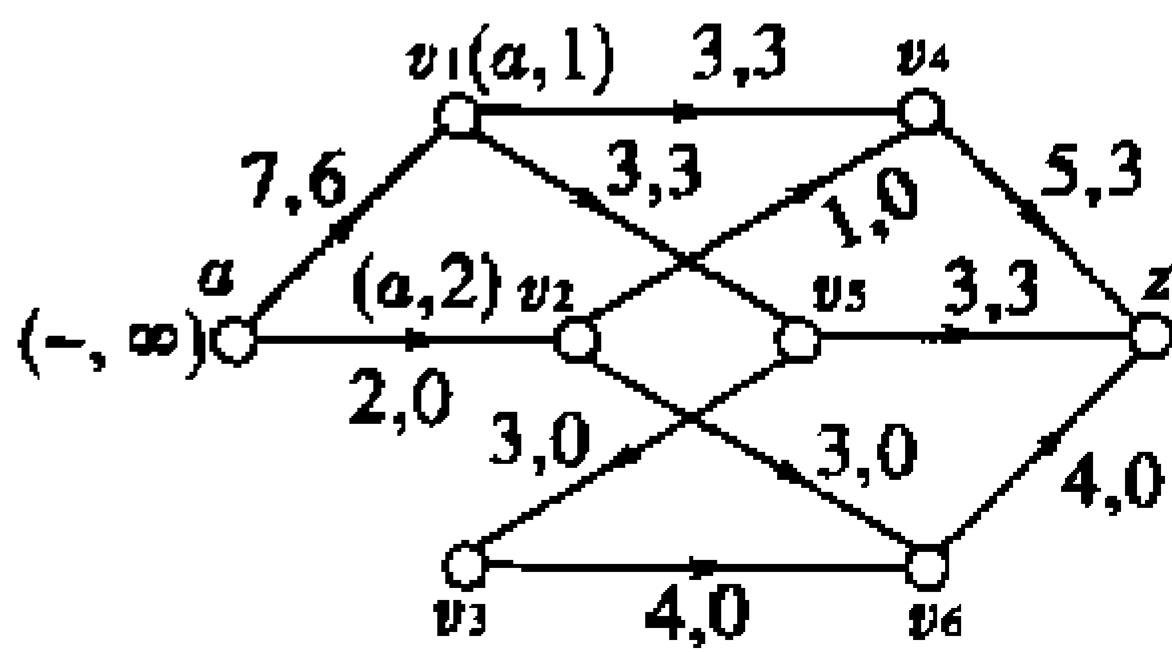
(6)其余标识同(5)



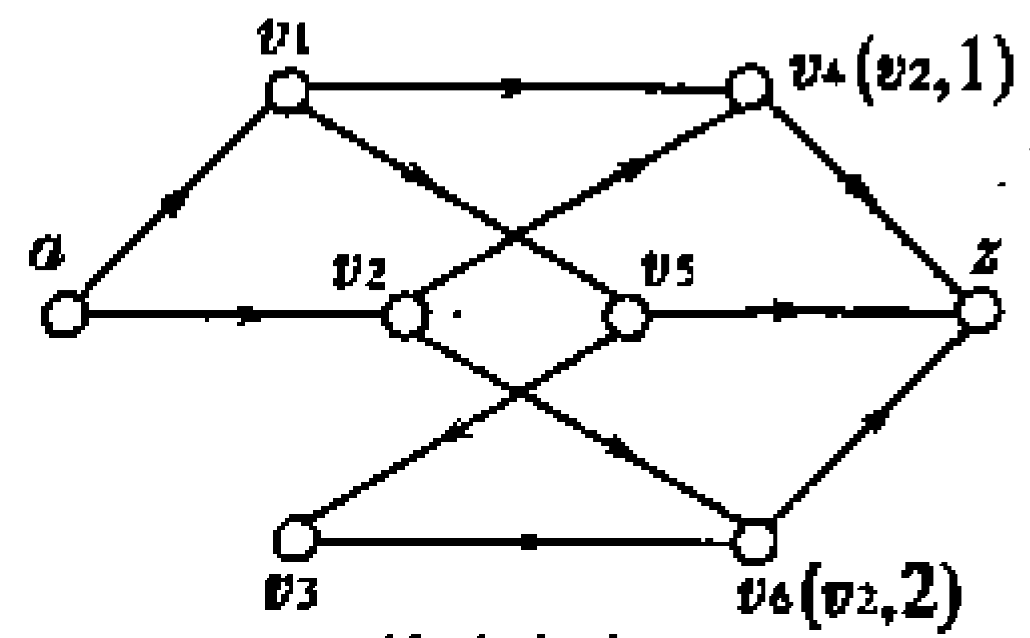
(7)其余标识同(6)



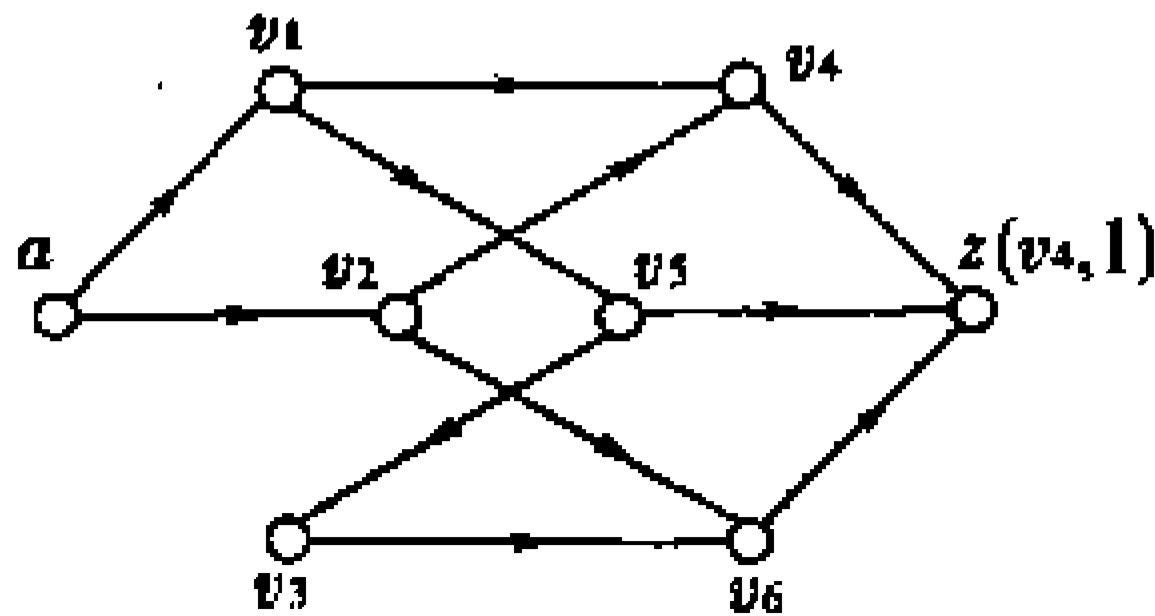
(8)其余标识同(7)



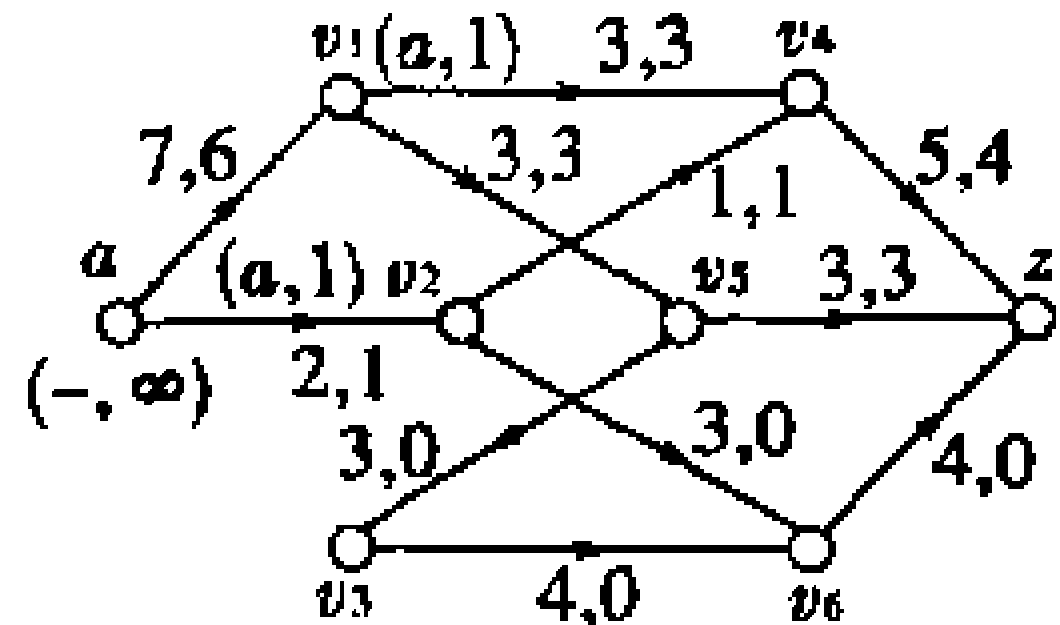
(9)$P=(a,v_1,v_5,z)$



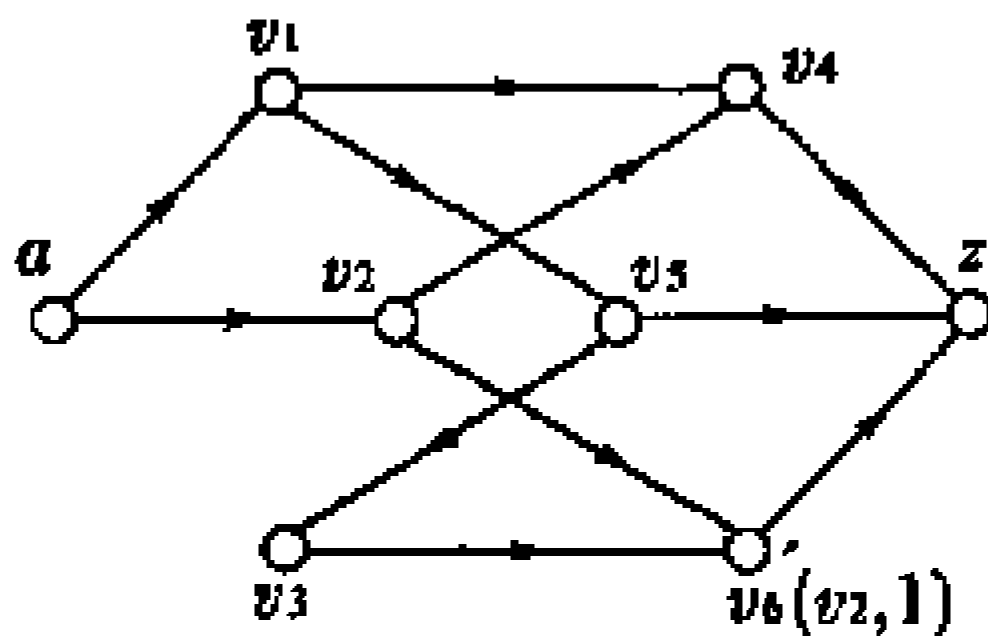
(10)其余标识同(9)



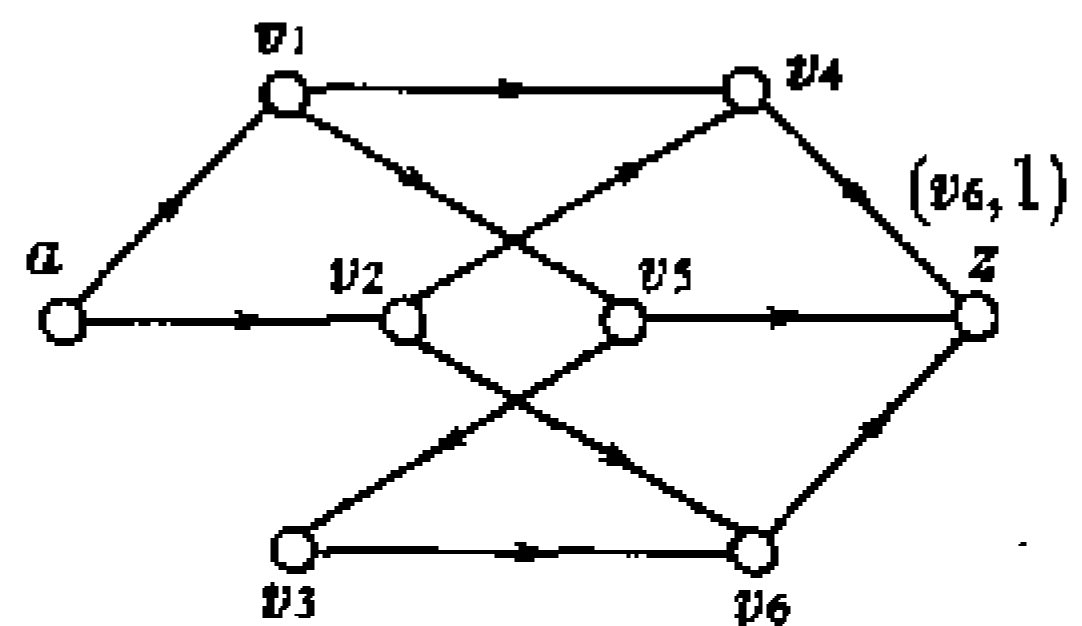
(11)其余标识同(10)



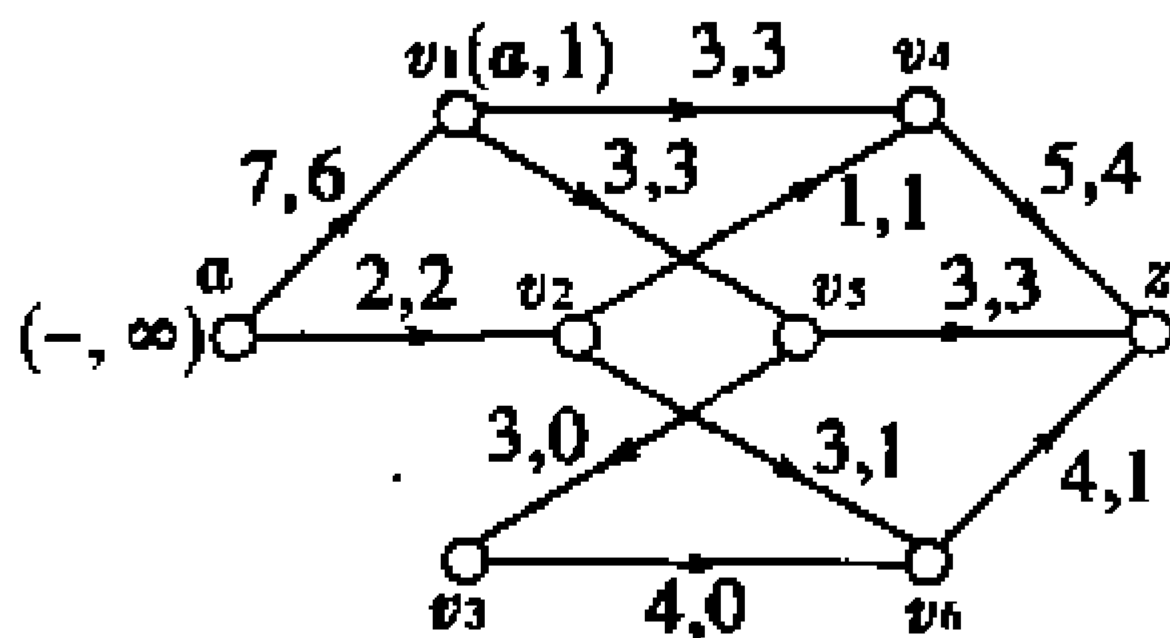
(12) $P=(a,v_2,v_4,z)$



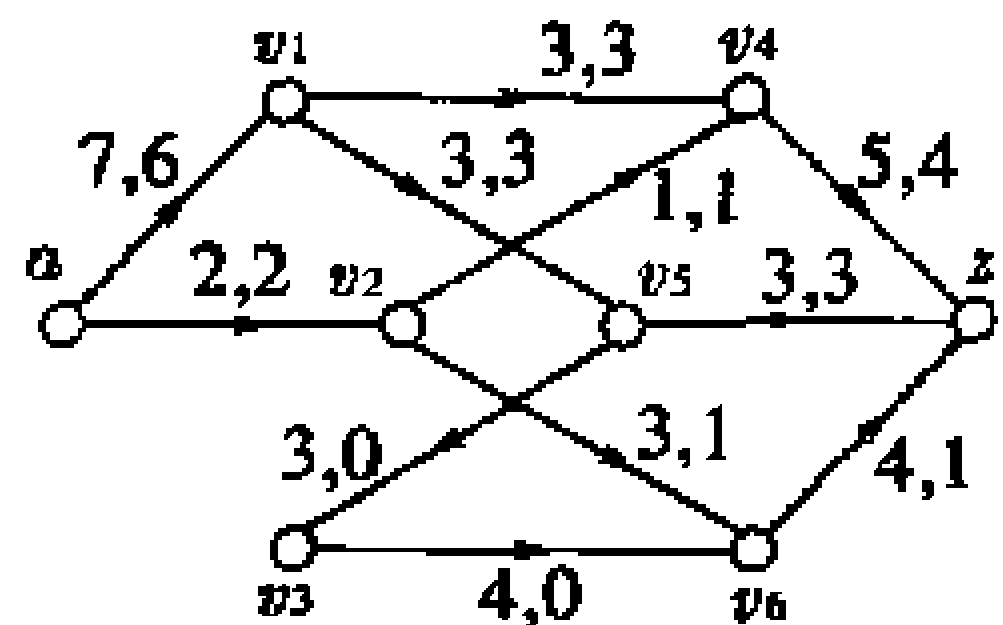
(13)其余标识同(12)



(14)其余标识同(13)



(15)$P=(a,v_2,v_6,z)$ $S=\phi$



(16)最大流 F

图 6-9

上述求网络最大流的算法又称为标号法(labelling method).在网络 G 中,假设关联于源 a 的边的容量之和是正整数 C.标号

法的每次迭代至少增加流的值为1,而流的值不会超过 C. 因此,经有限次步骤之后,该算法将终止.

定理3 标号法结束时生成网络 G 的一个最大流. 而且,如果 P 和 $\bar{P}$ 分别是标号法结束时已标记和没标记的顶点,则切割 $(P,\bar{P})$ 是最小的.

证明 设 P 和 $\bar{P}$ 分别为算法结束时已标记和没标记的顶点. 考虑边 $\langle i,j\rangle$,其中 $i\in P,j\in\bar{P}$,因为 i 已标号,我们必然有 $F_{ij}=C_{ij}$,否则,我们应对 j 进行标号. 现在考虑边 $\langle j,i\rangle$,其中 $j\in\bar{P}$,$i\in P$,因为 i 已标号,我们必然有 $F_{ij}=0$,否则,我们应对 j 进行标号. 由本章第一节的定理3知,算法结束时流最大而切割最小.

定理4(最大流最小切割定理) 在任何网络 G 中,最大流的值等于最小切割的容量.

证明 设 F 是网络 G 的最大流,由本节定理2知,G 中不存在 F 可增路. 如果 G 不包含 F 可增路,那么 G 存在最小切割 $(P,\bar{P})$,使得 $(P,\bar{P})$ 的每条边是 F 饱和的,而 $(\bar{P},P)$ 的每条边的流量是零. 因此,$\sum_{x\in P}\sum_{y\in\bar{P}}F_{xy}=C(P,\bar{P})$,且 $\sum_{x\in\bar{P}}\sum_{y\in P}F_{xy}=0$. 从而,$\mathrm{val}(F)=\sum_{x\in P}\sum_{y\in\bar{P}}F_{xy}-\sum_{x\in\bar{P}}\sum_{y\in P}F_{xy}=C(P,\bar{P})$.

流量为整数的最大流称为**整数最大流**. 由标号法很容易得到下列一条重要定理.

定理5(整数流定理) 任一网络中,若所有弧的容量都是整数,则必存在整数最大流.

证明 按标号法,其中运算是整数减法,整数中取最小值,所得的数仍是整数. 因此,只要初始可行流在各条边上的流量是整数,求得的最大流是整数最大流. 证毕

标号法是由 Ford 和 Fulk 在 1957 年首先提出的,然后由 Edmonds 和 Karp 在 1972 年稍加修正的.

标号法的基本思想是从网络 G 中任何一个已知流 F(如零流量)开始，递归地构作出一个其流量值不断增加的流序列，并且终止于最大流. 在每个新的流 F 作出之后，如果存在 F 可增路 P，则作出基于 P 的修正流 F'，然后将 F' 作为初始流重新执行算法. 如果不存在 F 可增路，则算法停止.

在算法的执行过程中，G 中顶点被分成三类：未被标号(即未曾进入 S)；已被标号但未被删去(即仍在 S 中)，已被标号且已被删去(即已从 S 中删去). 标号程序中应本着"先标号先删去"，即"先进先出"的原则，即在删去 i 之前应先删去在 i 之前已被标号过的顶点. 这样做的目的是确保选择一条最短 F 可增路. 例如，考虑图 6-10 所示的容量网络 G. 显然 G 中存在一个值为 $2m$ 的流 F. 若从零流量开始，交错选取 $(x, x_1, x_2, x_5, x_6, y)$ 和 $(x, x_4, x_5, x_2, x_3, y)$ 作为可增路，每次修正流的值增加单位值，需要进行 $2m+1$ 次标号过程. 由于 m 的任意性，因此算法不仅与网络规模，而且与 m 有关. 若按"先标号先删去"的原则，标号程序只需进行 2 次就可获得最大流.

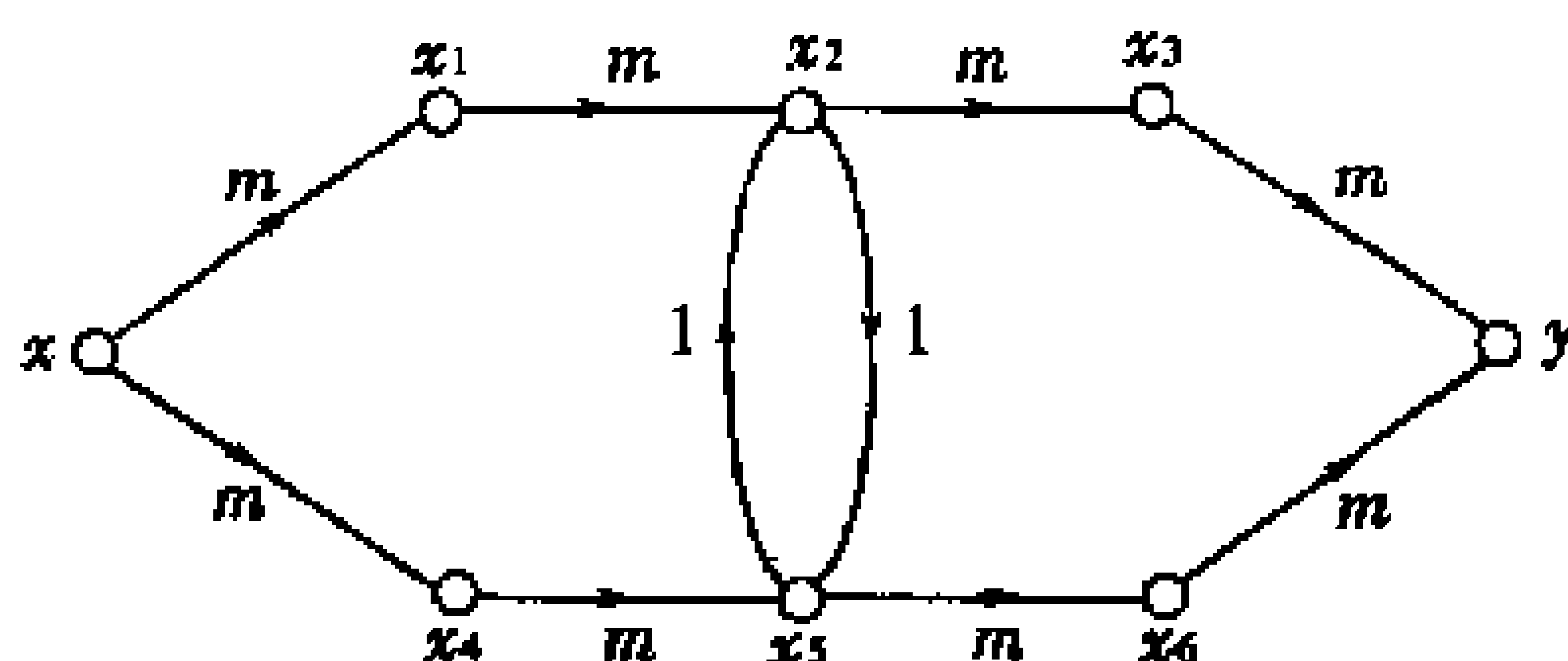


图 6-10

事实上，对于一般的整容量网络 G，我们可以证明：若每次修正流 F' 是通过最短 F 可增路获得的，那么标号程序至多进行 $\frac{1}{2}|E(G)||V(G)|$ 次就可以获得最大流.

值得指出的是，对无理数容量网络，Ford & Fulkerson 给出了

一个例子，说明标号法不能在有限步内停止，并且计算过程中得到的流序列也不收敛于最大流.

第三节　Menger 定理

最大流最小切割定理是图论中的一条重要定理.适当地选择网络，可容易地得到许多其它定理.著名的 Menger 定理也可以由它导出.为此，先给出有关概念.

设 x 和 y 是图 G 中不同两顶点.$L=\{l_1,l_2,\cdots,l_m\}$ 是 G 中连结 x 和 y 的路(若 G 是有向图，则指从 x 到 y 的有向路)的集合，若 L 中任何两条路 l_i 与 l_j 均有 $V(l_i)\cap V(l_j)=\{x,y\}$，则称 L 是 G 中**内部点不交的(internally vertex-disjoint)** xy 路的集合；若 L 中任何两条路 l_i 与 l_j 均有 $E(l_i)\cap E(l_j)=\varnothing$，则称 L 是 G 中**边不交的(edge-disjoint)** xy 路的集合.我们用 $\xi_G(x,y)$ 和 $\eta_G(x,y)$ 分别表示图 G 中内部点不交和边不交的 xy 路的最大条数.

x 和 y 是图 G 的两个不同顶点，$P\subseteq V(G)$，$\bar{P}=V(G)-P$，且 $x\in P$，$y\in\bar{P}$，$B=\{\{V_i,V_j\}\mid V_i\in P,V_j\in\bar{P}\}$(若 G 为有向图，则 $B=\{\langle V_i,V_j\rangle\mid V_i\in P,V_j\in\bar{P}\}$)称为$(P,\bar{P})$作用下的 G 的 xy **截边集(cut edge-set)**.具有最小边数的 xy 截边集称为**最小 xy 截边集(minimum cut edge-set)**.我们用 $\lambda_G(x,y)$ 表示 G 中最小 xy 截边集的边数.显然，$\eta_G(x,y)\leqslant\lambda_G(x,y)$.

引理 1　设 G 是以 x 为源 y 为汇的网络，每条边的容量为 1，则

(1) G 中最大流的值等于 G 中边不相交的从 x 到 y 的有向路的最大条数.

(2) G 中最小切割的容量等于 G 中最小 xy 截边集的边数.

证明 (1)由整数流定理,设 F 是 G 中的整数最大流. 在 G 中删去流量为零的边得 H,对于 H 中每条边 $\langle i,j\rangle$,均有 $F_{ij}=1$. 我们有

$$\begin{cases}\deg_H^-(x)-\deg_H^+(x)=\deg_H^+(y)-\deg_H^-(y)=\mathrm{val}(F)\\ \deg_H^-(i)=\deg_H^+(i) \quad i\neq x,y\end{cases}$$

因此,H 中存在 $\mathrm{val}(F)$ 条边不相交的从 x 到 y 的有向路,由于 $\eta_G(x,y)$ 的最大性,因而 $\mathrm{val}(F)\leqslant\eta_G(x,y)$.

设 $l_1,l_2,\cdots,l_m(m=\eta_G(x,y))$ 是 G 中 m 条边不相交的从 x 到 y 的有向路,令

$$F_{ij}^*=\begin{cases}1 & \langle i,j\rangle\in\bigcup_{i=1}^{m}l_i\\ 0 & 否则\end{cases}$$

得到 G 的一个可行流 F^*,且 $\mathrm{val}(F^*)=m$. 因为 F 是最大流,所以 $m=\mathrm{val}(F^*)\leqslant\mathrm{val}(F)$.

因此,$\mathrm{val}(F)=\eta_G(x,y)$.

(2)设 $(P,\overline{P})$ 是 G 的最小切割,B 是 $(P,\overline{P})$ 作用下的 G 的 xy 截边集,则 $G-B$ 中不存在从 x 到 y 的有向路,所以

$$\lambda_G(x,y)\leqslant|B|=C(P,\overline{P})$$

现设 B^* 是 G 中最小 xy 截边集,即 $|B^*|=\lambda_G(x,y)$. 令 $S=\{t\mid t\in V(G),G-B^*$ 中存在从 x 到 t 的有向路$\}$. 于是 $x\in S$, $y\notin S$. 显然 $(S,\overline{S})$ 是 G 的一个切割,且 $(S,\overline{S})$ 作用下的 G 的 xy 截边集记作 B'.

因此,$|B'|=C(S,\overline{S})\geqslant C(P,\overline{P})=|B|$.

又因为 $E(G-B^*)\cap B'=\varnothing$,否则 $G-B^*$ 中从 x 到 y 存在有向路,这与 B^* 的定义矛盾. 所以 $B'\subseteq B^*$,并且 $\lambda_G(x,y)=|B^*|\geqslant|B'|=C(S,\overline{S})\geqslant C(P,\overline{P})=|B|$.

因此,$\lambda_G(x,y)=C(P,\overline{P})$.

定理 1 设 x 和 y 是有向图 G 中任意两个不同顶点，则 G 中从 x 到 y 的边不相交的有向路的最大条数 $\eta_G(x,y)$ 等于 G 的最小 xy 截边集的边数 $\lambda_G(x,y)$.

证明 在有向图 G 的每条弧上给定单位容量. 因仅考虑从 x 到 y 的有向路，删去以 x 为终点的边和以 y 为起点的边. 得到一个以 x 为源，y 为汇的网络 D.

对于网络 D，由最大流最小切割定理知，存在最大流 F 和最小切割 $(P,\overline{P})$，且 $\mathrm{val}(F)=C(P,\overline{P})$. 而 $\eta_G(x,y)=\mathrm{val}(F)$，$C(P,\overline{P})=\lambda_G(x,y)$，所以

$$\eta_G(x,y)=\lambda_G(x,y)$$

定理 2 设 x 和 y 是连通无向图 G 中任两个不同顶点，则 $\eta_G(x,y)=\lambda_G(x,y)$.

证明 对连通无向图 G，构造 G 的对称有向图 D. 若 $l=(V_0,V_1,\cdots,V_n)$ 是 G 的一条路当且仅当 l 是 D 的一条有向路. G 中从 x 到 y 的边不相交的路的最大个数等于 D 中从 x 到 y 的边不相交的有向路的最大个数，同时，$(P,\overline{P})$ 作用下的 G 的最小 xy 截边集的边数等于 D 的最小 xy 截边集的边数.

以上两个定理是 Menger 定理的边形式，下面再给出 Menger 定理的点形式.

设 x 和 y 是图 G 的两个不同顶点. 若存在 $S\subseteq V(G)-\{x,y\}$ 使 $G-S$ 中不存在从 x 到 y 的路(若 G 是有向图，则是指从 x 到 y 的有向路)，则称 S 为 G 中 xy **分离集(separating set)**. 具有最小顶点数目的 xy 分离集称为**最小 xy 分离集**. G 中最小 xy 分离集的顶点数目记作 $K_G(x,y)$. 显然，$\xi_G(x,y)\leqslant K_G(x,y)$.

若 x 和 y 是图 G 中不相邻的两个顶点，则 G 中 xy 分离集一定存在.

定理 3 设 x 和 y 是图 G 中不相邻的两个顶点，则 $\xi_G(x,y)$

$=K_G(x,y)$.

定理的证明留作练习.

下面,再来讨论二分图的最大匹配问题.

设 G 是具有二部划分(V_1,V_2)的二分图,构造相应的以 a 为源 z 为汇的网络 N,N 的顶点集 $V(N)=V_1\cup V_2\cup\{a,z\}$,边集 $E(N)=\{\langle a,x\rangle\mid x\in V_1\}\cup\{\langle y,z\rangle\mid y\in V_2\}\cup\{\langle x,y\rangle\mid x\in V_1,y\in V_2\}$,在边$\langle a,x\rangle$和$\langle y,z\rangle$上都取容量为 1,在边$\langle x,y\rangle$上取容量为$\infty$.例如,图 6-11(b)即为 6-11(a)所示二分图的网络.

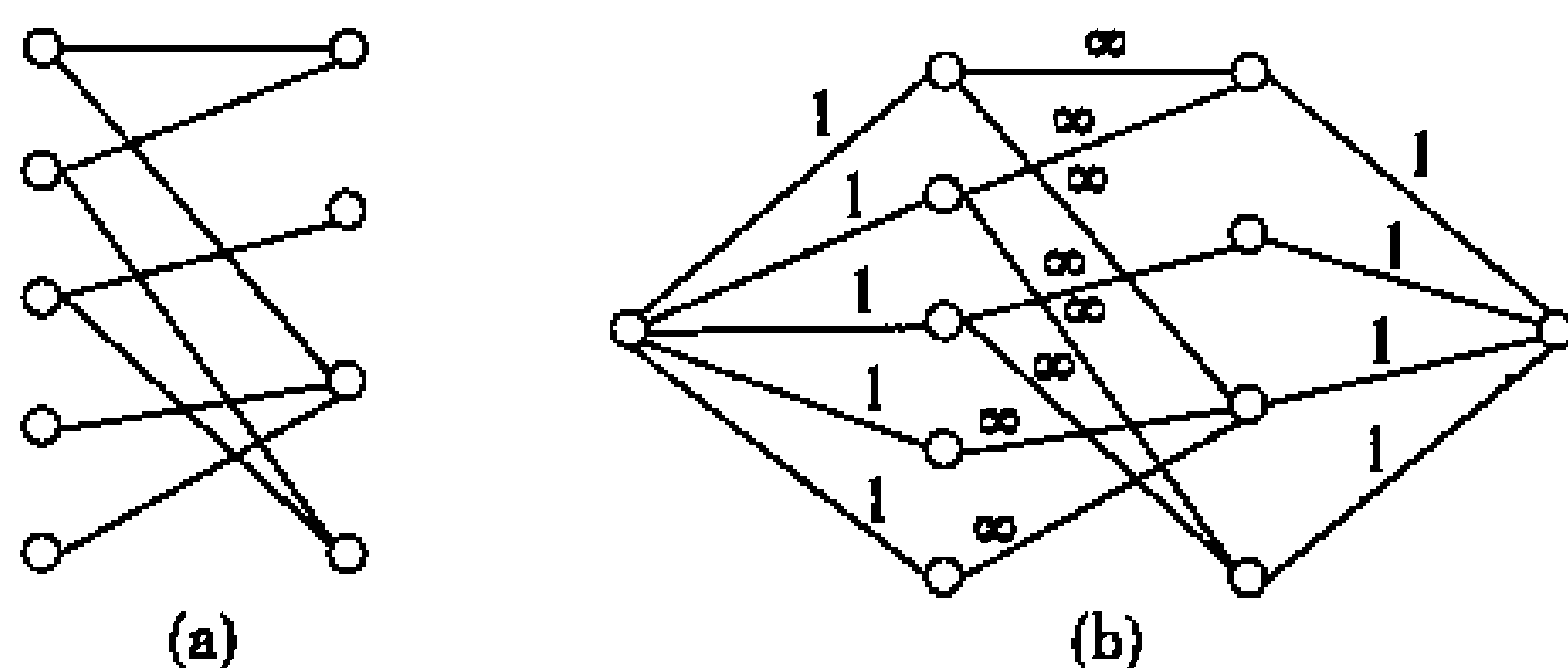


图 6-11

定理 4 设 G 是具有二部划分(V_1,V_2)的二分图,其相应的网络为 N,则

(1)N 的流给出 G 的一个匹配.顶点 $x\in V_1$ 与 $y\in V_2$ 匹配当且仅当边$\langle x,y\rangle$的流量为 1.

(2)N 的最大流对应于 G 的最大匹配.

证明 设 a 是网络 N 的源,z 为 N 的汇,并假定 N 有已知流.

(1)假定边$\langle x,y\rangle$的流量为 1,其中 $x\in V_1,y\in V_2$.进入顶点 x 的唯一边是$\langle a,x\rangle$.边$\langle a,x\rangle$必然有流量 1,于是进入 x 的流量为 1.因为流出 x 的流量也是 1,流量为 1 且以 x 为起点的唯一边为$\langle x,y\rangle$.类似地,流量为 1 且以 y 为终点的边只有边$\langle x,y\rangle$.如果 E 是流量为 1 的形如$\langle x,y\rangle$的边的集合,则 E 的成员没有公共

点，于是 M 是 G 的一个匹配.

(2)设 M 是 G 的最大匹配.在 G 的相应网络 N 中，对应于 M 的每一边 $\langle x,y\rangle$，有一条从 a 到 z 的有向路 $\langle a,x,y,z\rangle$，其中 a 为 N 的源，z 为 N 的汇.在 N 中定义流 F，在这种有向路的每边上的流量为 1，在其余边上的流量均为 0.由于 M 是匹配，上述有向路是内部点不相交的，因此 $\mathrm{val}(F)=|M|$.因而，对于网络 N 的最大流 F^*，显然有 $|M|=\mathrm{val}(F)\leqslant\mathrm{val}(F^*)$.

由于 F^* 是 N 的最大流，不妨设 F^* 是整数流.由 N 的容量知，F^* 在每条边的流量为 0 或 1.由于边 $\langle a,x\rangle$ 与 $\langle y,z\rangle$ 的容量为 1，且从 a 到 x 以及从 y 到 z 只有一条有向边，因此若 $F^*_{xy}=1$，则 $F^*_{xy'}=0$，$F^*_{x'y}=0$，其中 $x\neq x'$，$y\neq y'$.边集 $\{\langle x,y\rangle\mid F^*_{xy}=1\}$ 对应于 G 的一个匹配 M'，且 $|M'|=\mathrm{val}(F^*)$.又因 M 是最大匹配，因此，$\mathrm{val}(F^*)=|M'|\leqslant|M|$.

所以，$|M|=\mathrm{val}(F^*)$.

第四节　最小费用最大流

已经讨论了在网络中求最大流的方法，一般在网络中还有一个“费用”问题.一条边上不仅标上容量，还有单位流量的费用.下面我们先看一个具体实例.

某日旅行社要安排旅客从 A 城市坐飞机到 B 城市，途径若干个机场，有几种路径由 A 到 B.旅客即可结伴而行，也可单独旅行.问这家旅行社如何安排使运送的旅客最多而旅费最少？

我们构造一个网络 G，用顶点表示 A、B 以及它们之间的机场，其中 A 为源 B 为汇；用边表示机场之间将要起飞的各次班机.每条边的容量是对应班机的座位数，而每条边的费用是相应班机上一张机票的价格.问题转化成求网络 G 的最大流且使所花的费用最小.这就是我们要讨论的最小费用最大流问题.

设 G 是以 x 为源 y 为汇的网络，C 是 G 的容量，b 是 G 的单位流量费用，F 是 G 的流，则 $b(F)=\sum_{\langle i,j\rangle\in E(G)} F_{ij}b_{ij}$ 称为 F 的费用．最小费用最大流问题，就是求网络 G 的最大流 F 且使费用 $b(F)$ 最小．这样的流称为**最小费用最大流**（**minimum-cost maximum flow**）．P 是 G 中有指定正向的圈，令

$$\sigma_{ij}=\begin{cases}C_{ij}-F_{ij} & \langle i,j\rangle\text{是 } P \text{ 中正向边}\\ F_{ij} & \langle i,j\rangle\text{是 } P \text{ 中反向边}\end{cases}$$

$$\sigma(P)=\min\{\sigma_{ij}\mid\langle i,j\rangle\in E(P)\}$$

若 G 中的圈 P 存在一个定向使 $\sigma_{ij}>0$，则称 P 为 F **增广圈**（**increment cycle**）．对于 F 增广圈 P，我们可以定义 F^*：

$$F_{ij}^*=\begin{cases}F_{ij}+\sigma(P) & \langle i,j\rangle\text{为 } P \text{ 的正向边}\\ F_{ij}-\sigma(P) & \langle i,j\rangle\text{为 } P \text{ 的负向边}\\ F_{ij} & \text{其它}\end{cases}$$

容易验证，F^* 是 G 的流且 $\mathrm{val}(F^*)=\mathrm{val}(F)$．$F^*$ 称为基于 F 增广圈 P 的**修正流**（**revised flow**）．

设 P 是 F 增广圈，则定义 $b(P,F)=\sum_{\langle i,j\rangle\in P^+} b_{ij}-\sum_{\langle i,j\rangle\in P^-} b_{ij}$，其中 P^+ 为 P 的正向边集，P^- 为 P 的负向边集，为 F 增广圈 P 的费用．若对 G 中其流量值等于 $\mathrm{val}(F)$ 的任何一个流 F' 均有 $b(F)\leqslant b(F')$，则称 F 为**最小费用流**．

对于任意顶点 j，我们称流 F 在顶点 j 的净流出量 $\sum_i F_{ji}-\sum_i F_{ij}$ 为顶点 j 的流量．设 F 和 F' 是两个在每个顶点的流量相等的流，如果在圈 P 上满足

$$\begin{cases}F_{ij}<F_{ij}' & \langle i,j\rangle\in P^+\\ F_{ij}>F_{ij}' & \langle i,j\rangle\in P^-\end{cases}$$

则称在圈 P 上 F' 优于 F，记作 $F<F'$. 当然，这种"优于"关系是相对于圈 P 的某个定向而言的.

引理 1 如果 F, F' 是两个在每个顶点的流量都相等的不同流，则存在圈 C 使得在 C 上 $F<F'$.

证明 由已知 $F\neq F'$，则存在边 $\langle i,j\rangle$ 使得 $F_{ij}\neq F'_{ij}$，不妨设 $F_{ij}<F'_{ij}$. 考虑顶点 j，已知 F, F' 在顶点 j 的流量相等，则存在边 $\langle j,t\rangle$ 使 $F_{jt}<F'_{jt}$ 或者存在边 $\langle t,j\rangle$ 使 $F_{tj}>F'_{tj}$. 再对顶点 t 进行类似讨论，由顶点个数的有限性，故必存在圈 P 使得对 P 的任意边 e, $F_e\neq F'_e$. 由 P 的构造方法，存在 P 的定向使在 P 上 $F<F'$.

引理 2 如 F, F' 是两个在每点流量相等的流，则从 F 经有限次在圈上的调整可以得到 F'.

证明 令 $m=|\{e|F_e\neq F'_e\}|$，如果 $m=0$，则 $F=F'$，结论成立，不妨设 $m\neq 0$，由引理 1 知，存在圈 P 使得 $F<F'$. 令 $F^{(1)}$ 是 F 在圈 P 上的调整，且调整值 $\varepsilon_1=\min\{|F_e-F'_e||e\in E(P)\}$，则 $m_1=|\{e|F^{(1)}_e\neq F'_e\}|\leqslant m-1$. 重复上述过程，在第 t 次有 $m_t=0$，即 $F^{(t)}=F'$.

由引理 2 的证明知，对任意边 e，如果 $F_e<F'_e$，则 $F_e\leqslant F^{(1)}_e\leqslant F^{(2)}_e\leqslant\cdots\leqslant F^{(t)}_e=F'_e$；如果 $Fe>F'_e$，则 $F_e\geqslant F^{(1)}_e\geqslant F^{(2)}_e\geqslant\cdots\geqslant F^{(t)}_e=F'_e$.

定理 1 G 是以 x 为源 y 为汇的网络，F 是最小费用流 $\Leftrightarrow G$ 中每条 F 增广圈 P 都有 $b(P,F)\geqslant 0$.

证明 ($\Rightarrow$) 设 F 是最小费用流，P 是 F 增广圈，并设 F' 为基于 F 增广圈 P 的修正流. 由于 $\mathrm{val}(F)=\mathrm{val}(F')$，所以

$$0\leqslant b(F)-b(F')=\sigma(P)\Big(\sum_{\langle i,j\rangle\in P^{+}}b_{ij}-\sum_{\langle i,j\rangle\in P^{-}}b_{ij}\Big)$$

$$=\sigma(P)b(P,F)$$

由于 $\sigma(P)>0$,所以 $b(P,F)\geqslant 0$.

($\Leftarrow$)设 F^* 是网络 G 的任意流,由引理 2 知,从 F 经有限次调整后可以得到F^*,设依次在圈 $P_1,P_2,\cdots,P_t$ 上进行调整,从 F 得到 F^*.

对 P_1 上的任意边 e,当 $e\in P_1^+$ 时,$F_e<F_e^*\leqslant c(e)$,而当 $e\in P_1^-$ 时,$F_e>F_e^*\geqslant 0$,其中 c 是G 的容量,于是 P_1 是 F 增广圈,由已知有 $b(P_1,F)\geqslant 0$.

令 $F^{(1)}$是 F 在P_1 上作 $\varepsilon_1=\min\left\{|F_e-F_e^*|\,\middle|\,e\in E(P_1)\right\}$的调整所得到的流,且 $b(F^{(1)})=b(F)+\varepsilon_1 b(P_1,F)\geqslant b(F)$. 接下去考虑圈 P_2,由于 P_2 上 $F^{(1)}<F^*$,即当 $e\in P_2^+$ 时,$F_e^{(1)}<F_e^*$;而当 $e\in P_2^-$ 时,$F_e^{(1)}>F_e^*$,于是当 $e\in P_2^+$ 时,$F_e\leqslant F_e^{(1)}<F_e^*\leqslant c(e)$;而当 $e\in P_2^-$ 时,$Fe\geqslant F_e^{(1)}>F_e^*\geqslant 0$,于是 P_2 也是 F 增广圈,故 $b(P_2,F)\geqslant 0$.

类似地可得 $b(F^{(2)})\geqslant b(F^{(1)})$,重复上述讨论,得到 $b(F)\leqslant b(F^{(1)})\leqslant b(F^{(2)})\leqslant\cdots\leqslant b(F^{(t)})=b(F^*)$,于是 F 是网络G 的最小费用流.

于是定理证毕.

定理 1 提供了解最小费用最大流问题的一个算法. 这个算法是 Klein(1967)提出的. 算法的基本思想是:已知 G 是以 x 为源 y 为汇的网络,从 G 的任何一个最大流 F 出发,检查每个 F 增广圈. 若所有 F 增广圈的费用都是非负的,则 F 就是所求的最小费用最大流. 若存在 F 增广圈 P 使得 $b(P,F)<0$,则用修正流 F^* 代替 F,再重复上述过程.

由此可以看出,从一个最大流 F 过渡到另一个最大流 F^* 的关键是寻找其费用为负的 F 增广圈. 我们可以通过作网络 G 的辅助图D 来实现这一点.

已知 G 的容量为C,流为 F,单位流量费用为 b. 设边 $e\in$

$E(G)$,用 $\tilde{e}$ 表示通过改变 e 的方向而得到的新边. 定义加权图 D 如下:

$$V(D)=V(G)$$

设边 $e\in E(G)$

若 $F_e=0$,则 $e\in E(D)$且 $\omega_e=b_e$;

若 $F_e=C_e$,则 $\tilde{e}\in E(D)$,且 $\omega_{\tilde{e}}=-b_e$;

若 $0<F_e<C_e$,则 $e,\tilde{e}\in E(D)$,$\omega_e=b_e$,$\omega_{\tilde{e}}=-b_e$.

设 P 是 G 中 F 增广圈,则对任何 $e\in E(P)$,均有 $0<F_e<C_e$. 因而 $e,\tilde{e}\in E(D)$. 设 $\overleftarrow{P}$ 是 D 中对应于 P 的有向圈使 $P^+\subseteq E(\overleftarrow{P})$. 令 $\overleftarrow{P}^+=P^+$,$\overleftarrow{P}^-=E(\overleftarrow{P})-\overleftarrow{P}^+$,则 P 的费用

$$\begin{aligned}b(P,F)&=\sum_{e\in P^+}b_e-\sum_{e\in P^-}b_e\\&=\sum_{e\in\overleftarrow{P}^+}\omega_e-\sum_{\tilde{e}\in\overleftarrow{P}^-}(-\omega_{\tilde{e}})\\&=\sum_{e\in E(\overleftarrow{P})}\omega_e=\omega_{\overleftarrow{P}}\end{aligned}$$

反之,设 $\overleftarrow{P}$ 是 D 中有向圈,令

$$\overleftarrow{P}^+=\{e\mid e\in E(\overleftarrow{P})\text{且 }\omega_e>0\}$$

$$\overleftarrow{P}^-=E(\overleftarrow{P})-\overleftarrow{P}^+$$

则 $\overleftarrow{P}^+\subset E(D)$且 $\overleftarrow{P}^-$ 中边全为 $\tilde{e}$. 令

$$P^+=\overleftarrow{P}^+,P^-=\{e\mid e\in E(D)\text{且 }\tilde{e}\in\overleftarrow{P}^-\}$$

则容易验证 $P^+\cup P^-$ 构成 D 中 F 增广圈,并且

$$\begin{aligned}\omega_{\overleftarrow{P}}&=\sum_{e\in E(\overleftarrow{P})}\omega_e=\sum_{e\in\overleftarrow{P}^+}\omega_e=\sum_{\tilde{e}\in\overleftarrow{P}^-}(-\omega_{\tilde{e}})\\&=\sum_{e\in P^+}b_e-\sum_{e\in P^-}b_e=b(P,F)\end{aligned}$$

于是寻找 G 中其费用<0 的 F 增广圈就等价于在 D 中寻找负权和的有向圈. 此有向圈称为负圈. 这样我们得到下面的定理.

定理 2 G 是以 x 为源 y 为汇的网络,D 是对应于网络 G 的

加权图，则 F 是 G 的最小费用流 $\Leftrightarrow D$ 中不含负圈.

算法　Klein 算法

(1)求网络 G 中的最大流 F；

(2)构造对应于 G 的加权图 D；

(3)求 D 中的负圈. 若无负圈，则停止，此时 F 是最小费用最大流. 若 D 含负圈 $\overleftarrow{P}$，则 $P^+ \cup P^-$ 是 F 增广圈(其正向与 $\overleftarrow{P}$ 的方向一致)，作修正流 F^*，并用 F^* 代替 F 转第(1)步.

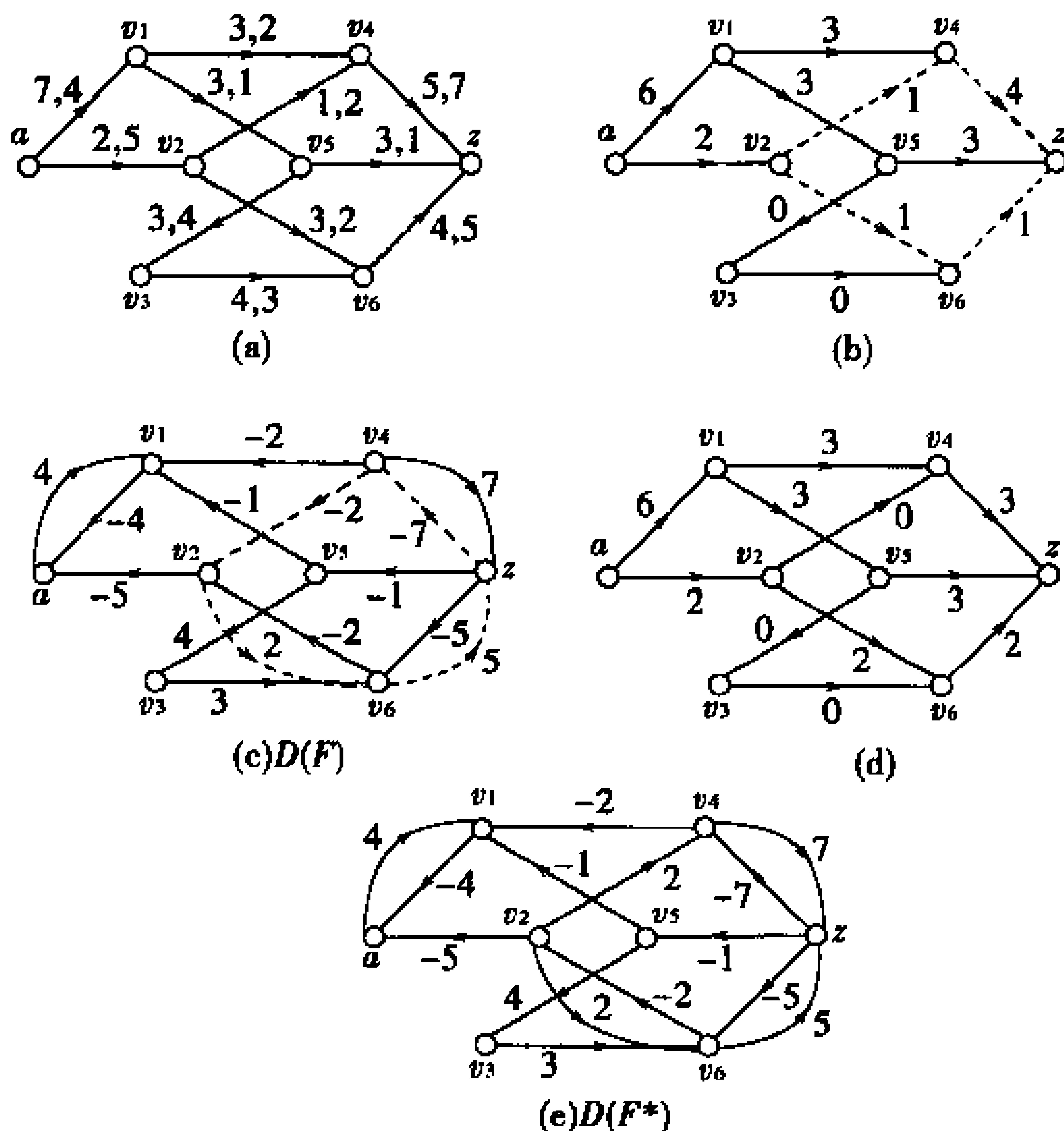


图 6-12

例 1　考察图 6-12(a)所示网络 G，其中边上的有序数对分别

为该边的容量和费用.我们已求得最大流 F 如图 6-12(b)所示.由 Klein 算法构作对应于网络 G 的加权图 $D(F)$ 如图 6-12(c)所示,图 6-12(c)中的虚线表示的是一条负圈,对应的 F 增广圈如图 6-12(b)中虚线所示.基于这个 F 增广圈的修正流 F^* 如图 6-12(d)所示.再执行 Klein 算法构作 $D(F^*)$ 如图 6-12(e)所示,$D(F^*)$ 中无负圈,所以 F^* 是最小费用最大流.事实上,$b(F)=83>81=b(F^*)$.

第五节　Petri 网简介

1962 年,Petri 在其博士论文中首先提出了 Petri 网,随着其在计算机科学、管理科学、系统分析等领域的广泛应用,Petri 网在国内外获得了广泛的重视.我们知道,前面所谈到的图论方法,充分显示了其表示静态关系的作用,而 Petri 网是揭示系统的动态特性,描述的是动态现象的关系.同时,Petri 网也是并行处理的图形模型,是刻划和研究并行处理的一种方法.

定义 1　有向图 $G=\langle V,E\rangle$,其中 $V=P\cup T$,且 $P\cap T=\emptyset$,E 中每条边与 P 中的一个顶点和 T 中的一个顶点相关联,称该有向图 $G=\langle V,E\rangle$ 为 **Petri 网**.其中 P 称为**位置集合**,T 称为**转移集合**.

事实上,Petri 网是一个有向二分图,两个不相交的顶点集被称作位置和转移.一般地,在 Petri 网中允许平行边,为简化起见,在本节不允许出现平行边.

一般地,用圆圈表示位置,用方框表示转移.

例 1　如图 6-13 所示是一个 Petri 网.

利用 Petri 网还可表示并行算法.例如求 $ab+cd$ 可用图 6-14 来表示:即 $a\times b$ 和 $c\times d$ 可在同一时刻进行计算.

定义 2　对一个 Petri 网分配给每个位置 P 一个非负整数 n,

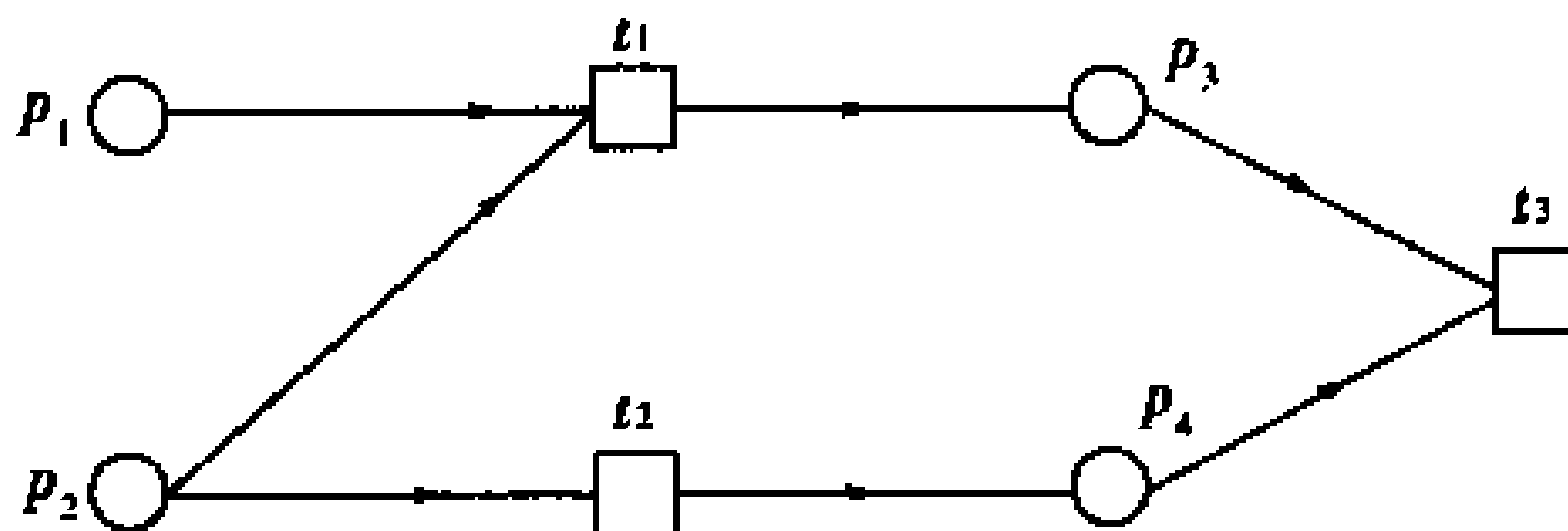


图 6-13　Petri 网示例

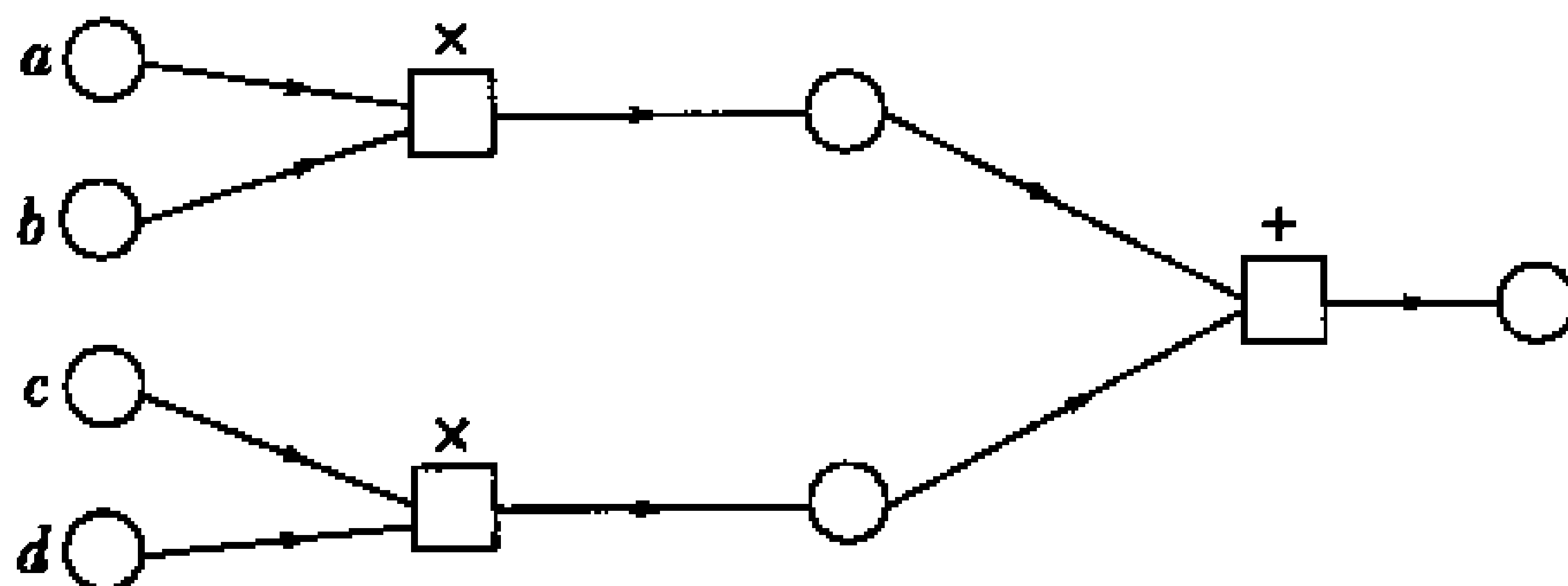


图 6-14

称这样的 Petri 网为**标记 Petri 网**.

一个 Petri 网的标记 M,可以看作是从位置集到非负整数集的一个函数.若标记 M 对位置点 P 赋值非负整数 n,我们说位置点 P 有 n 个记号,每个记号用一个黑点表示,画在圆圈的内部.

例 2　如图 6-15 所示是一个标记 Petri 网.

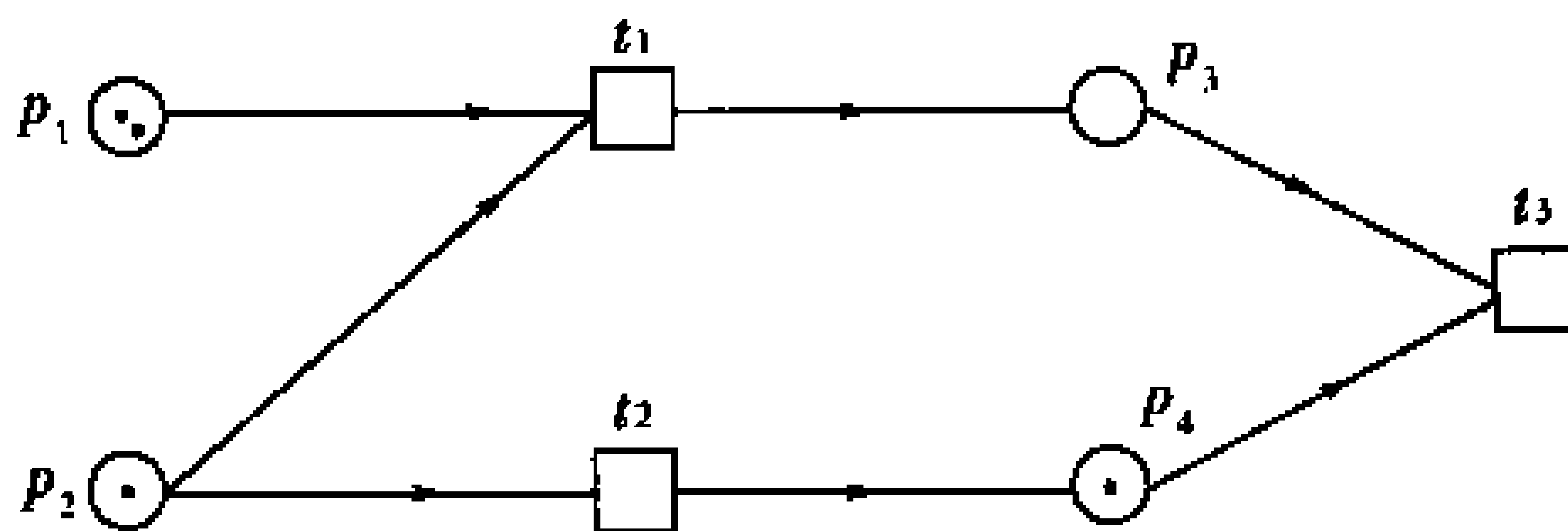


图 6-15

在模型中,若将位置点看作是条件,而转移点是执行或运算或事件,那么记号数则用以刻划条件是否成熟.

例 3　用转移点表示指令,而位置点表示一条指令能够被执

行的条件,图 6-16 显示了下列简单计算机程序的 Petri 网模型.

$$a=0$$
$$b=1$$
$$c=a+1$$
$$d=b+a$$
$$e=c+d$$

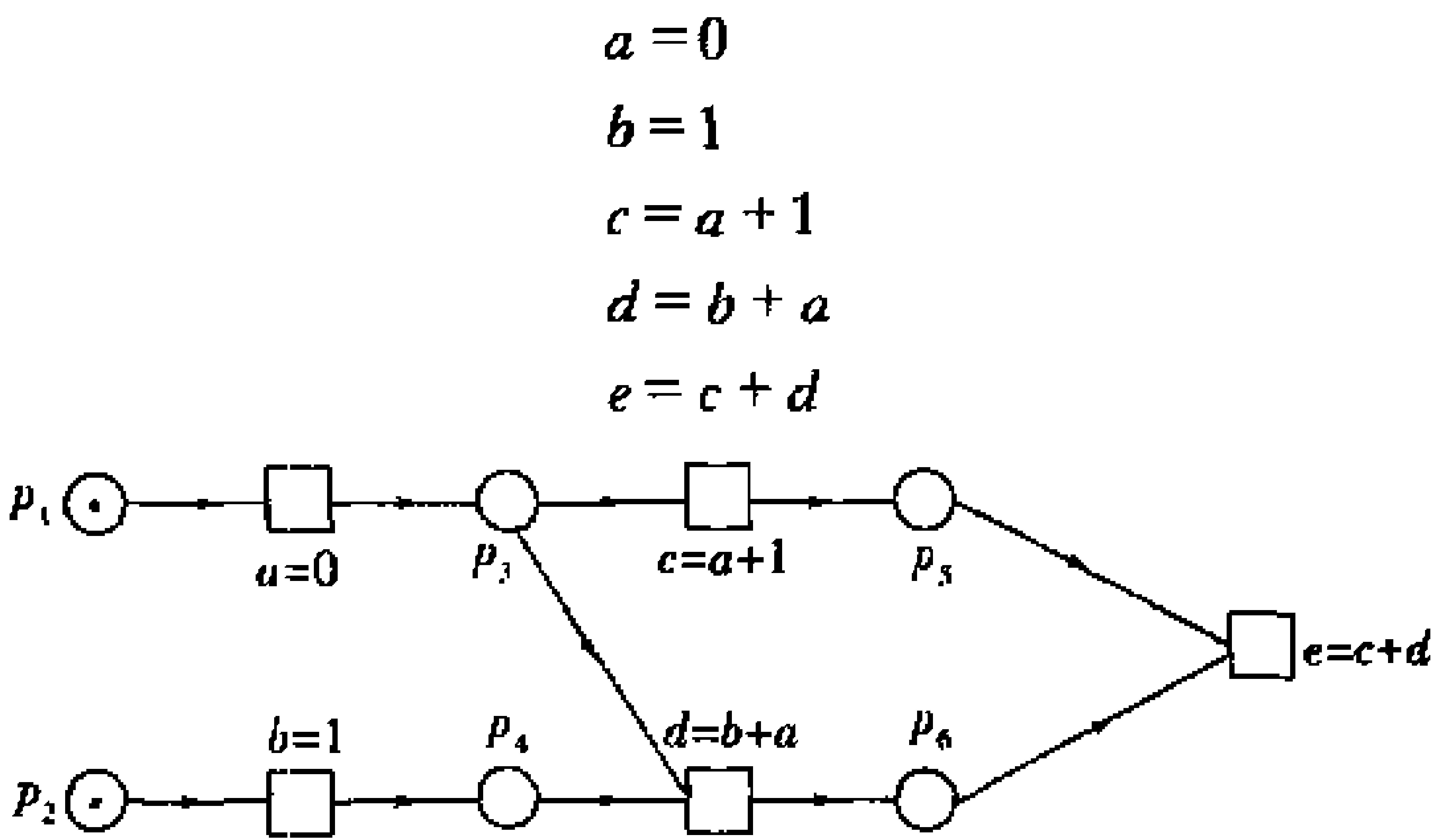


图 6-16

图中的记号说明执行 $a=0$ 和 $b=1$ 的条件满足.

定义 3 在一个 Petri 网中,如果一条边从位置点 P 指向转移点 t,称 P 是转移 t 的**输入位置**.类似地,若一条边从转移点 t 指向位置点 P,称 P 是转移 t 的**输出位置**.对一个转移点 t,如果每个输入位置至少有一个记号,称 t 是**使能的**.对于一个使能的转移,从每个输入位置中移出一个记号而在每个输出位置加入一个记号,则称**激发**了使能转移.

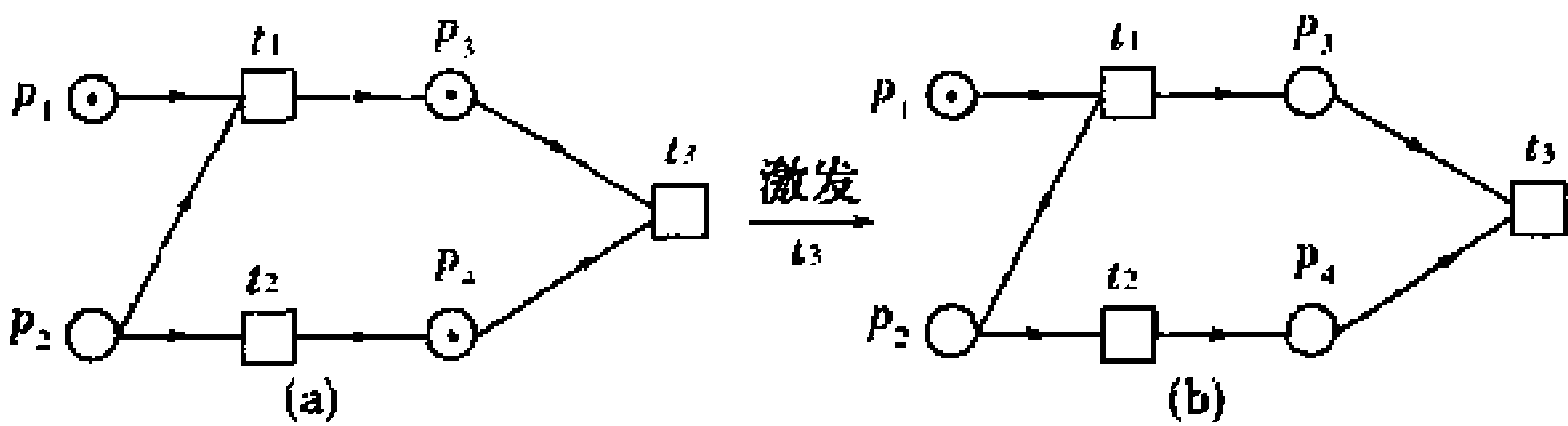


图 6-17

例 4 在图 6-15 的 Petri 网中,位置 P_1 和 P_2 是转移 t_1 的输入位置.转移 t_1 和 t_2 是使能的,但转移 t_3 不是使能的.如果我们

激发转移 t_1，得到图 6-17(a)所示的标记 Petri 网. 此时转移 t_3 是使能的. 如果我们激发 t_3，得到如图 6-17(b)所示的 Petri 网. 这时，无转移是使能的，于是无转移可激发.

定义 4 如果一个激发序列将一个标记 M 转换为标记 M'，称 M' 是从 M 可达的.

例 5 在图 6-18 中，M'' 通过首先激发 t_1 再激发 t_2 是从 M 可达的.

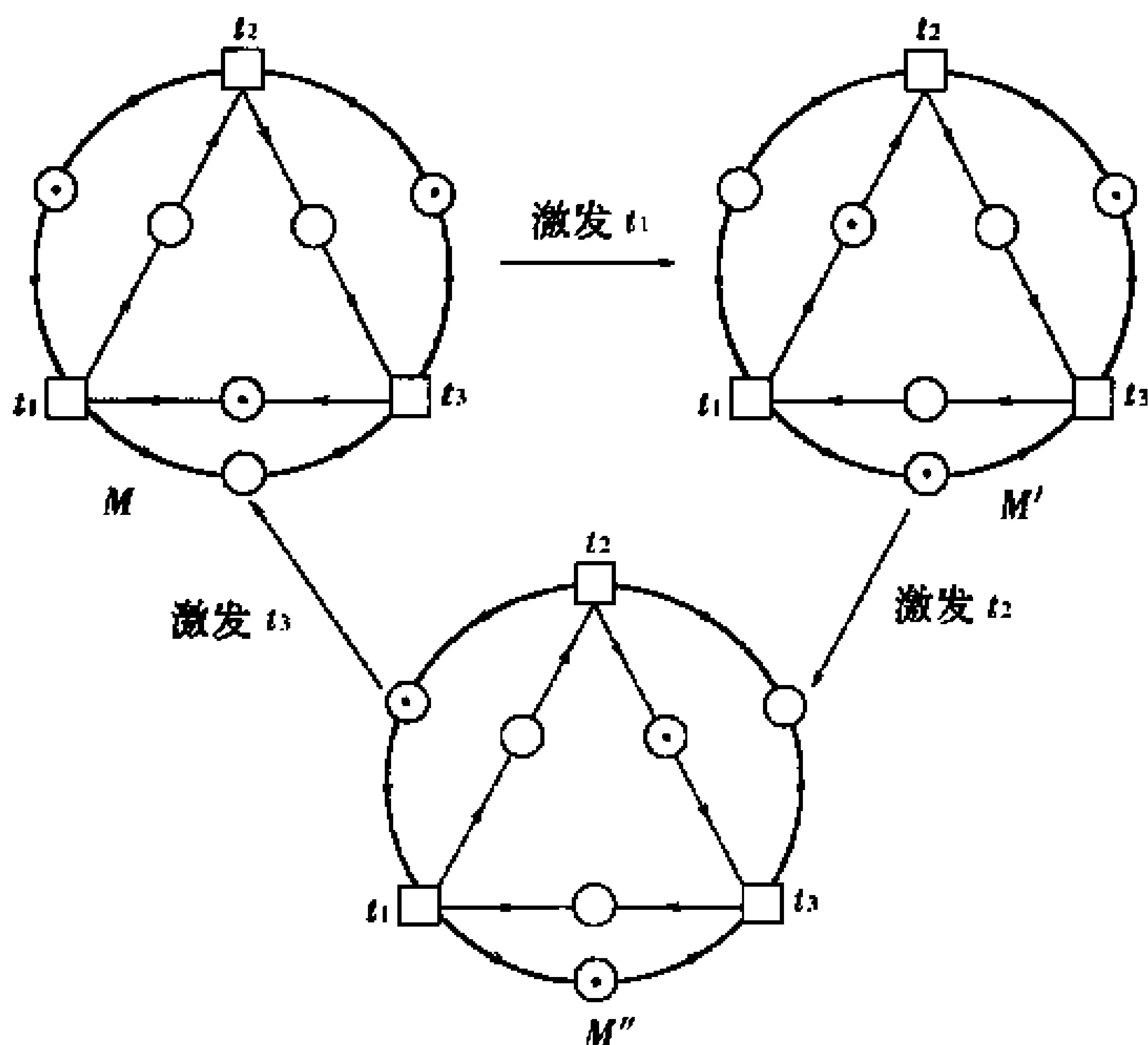


图 6-18

在 Petri 网建模中，激发转移将模拟事件的出现. 仅当一个事件执行的所有条件都能满足时，它才能够发生. 即仅当转移是使能的，它才能被激发. 在图 6-16 中，通过在位置 P_1 和 P_2 设置记号，标明了执行指令 $a=0$ 和 $b=1$ 是使能的，它们能够以任意次序或

同时地被激发. 仅当位置 P_3 和 P_4 有记号时,转移 $d=b+a$ 才是使能的. 但是,这些位置仅当转移 $a=0$ 和 $b=1$ 被激发后才会有记号. 也就是说,事件 $d=b+a$ 能够发生的条件是 $a=0$ 和 $b=1$ 必须被执行过. 从而可建立如图 6-16 所示的合法执行序列和相应的并行性模型.

在 Petri 网理论研究中,最重要的性质是**活动性**和**安全性**. 活动性与死锁的存在有关,安全性与有界存储容量有关.

下面通过一个实例来抽象出相应的基本概念.

例 6 甲乙两人共享具有一个磁盘驱动器 D 和一个打印机 P 的计算机系统. 每人既需要 D 又需要 P. 这时一种可能的 Petri 网模型如图 6-19 所示.

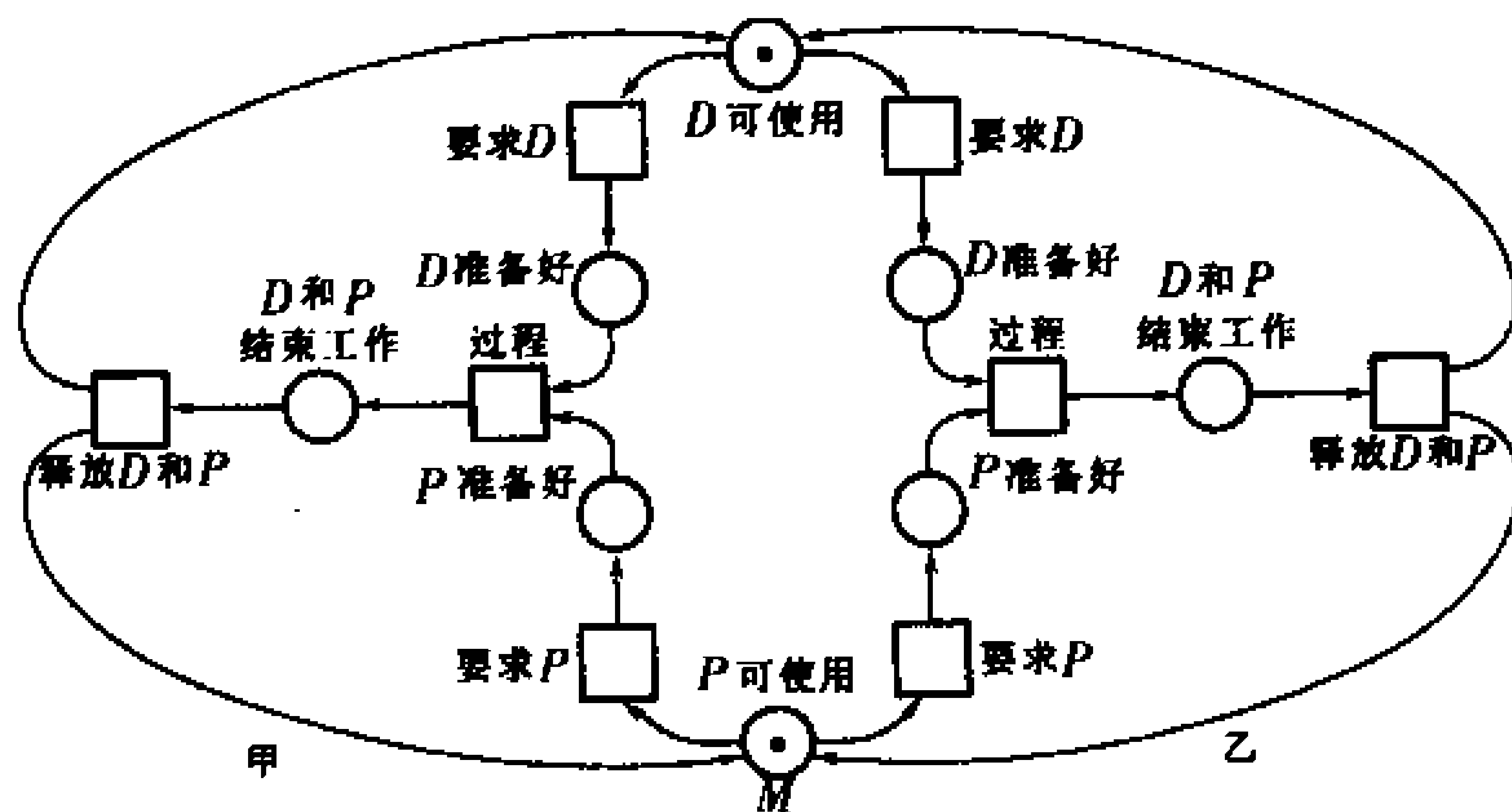


图 6-19

图 6-19 标记 M 指明 D 和 P 都是可用的.

现在假定甲要求 D,再要求 P(而乙不要求). 甲首先启动转移"要求 D",之后再启动转移"要求 P"以模拟这些事件的发生,得到图 6-20 所示的 Petri 网. 当甲完成过程并释放 D 和 P,将返回如图 6-19 所示的 Petri 网(其中,通过启动转移"过程",并启动转移"释放 D 和 P"来完成). 如果乙要求 D 且再要求 P,(而甲对 D

和P均不要求),可得到类似的启动序列.

若假定对图6-19所示的Petri网中,当甲要求D,而同时乙要求P,模拟事件发生并启动了适当的转移后,可得如图6-21所示的Petri网.注意这时再没有可启动的转移.甲正等待乙释放P,而乙正等待甲释放D.系统内部的活动停止,可认为发生了死锁.形象地说,如果一个有标记的Petri网中没有转移可启动,则称该有标记的Petri网是死锁的.在并行处理环境中,死锁的防止是极为重要的课题.

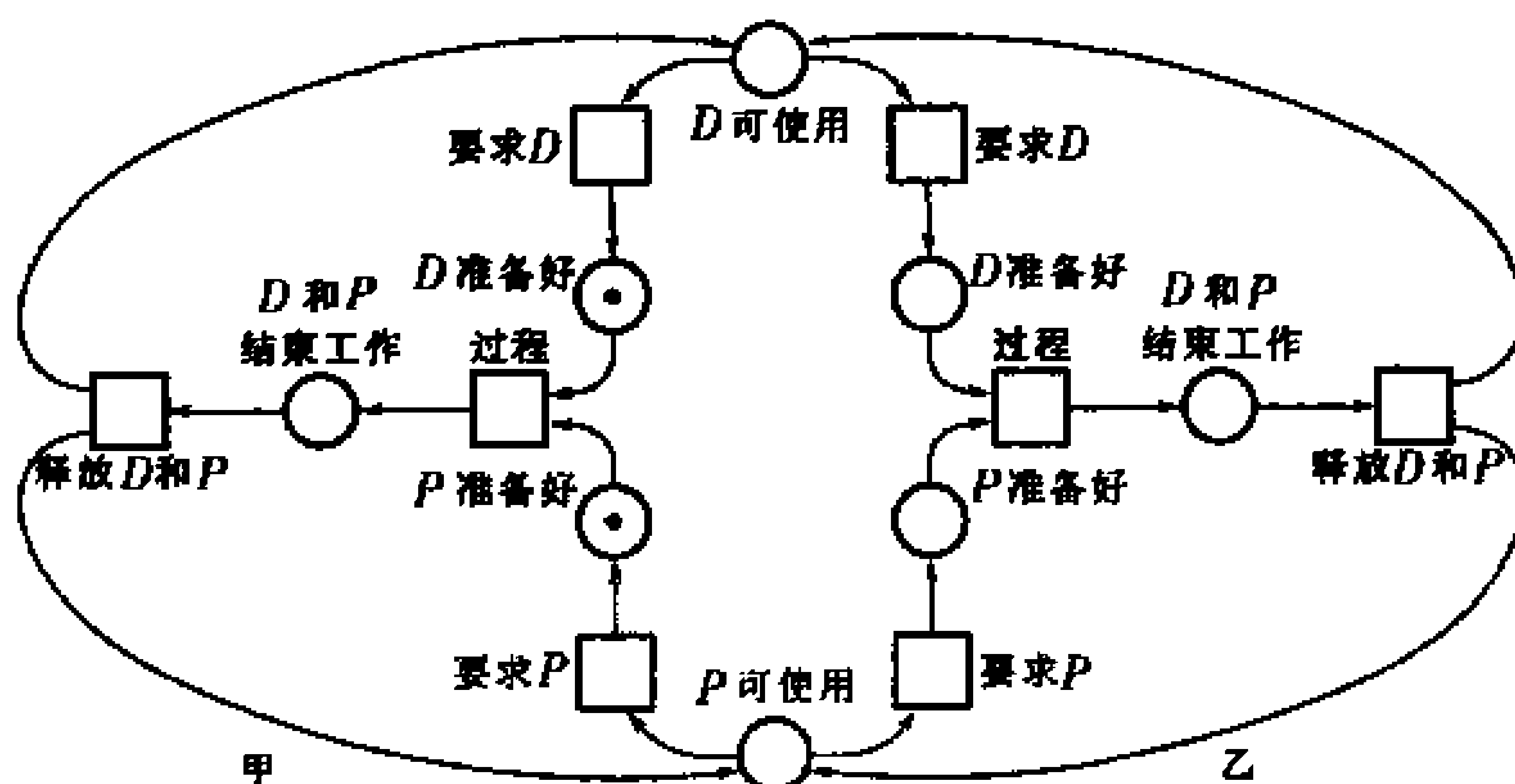


图6-20

如图6-20所示说明了甲启动“要求D和要求P”(乙不要求D和P)的Petri网.

图6-21为甲启动“要求D”和乙启动“”要求P”后的Petri网.由于甲、乙互不释放已有资源又申请未占用资源,导致Petri网死锁.

定义5 如果从Petri网的一个标记M开始,不论出现任何启动序列,都可能通过某些附加的启动序列的处理,启动任意给定的转移,则称M是活动的.

如果一个标记M对于某个Petri网P是活动的,则不论启动

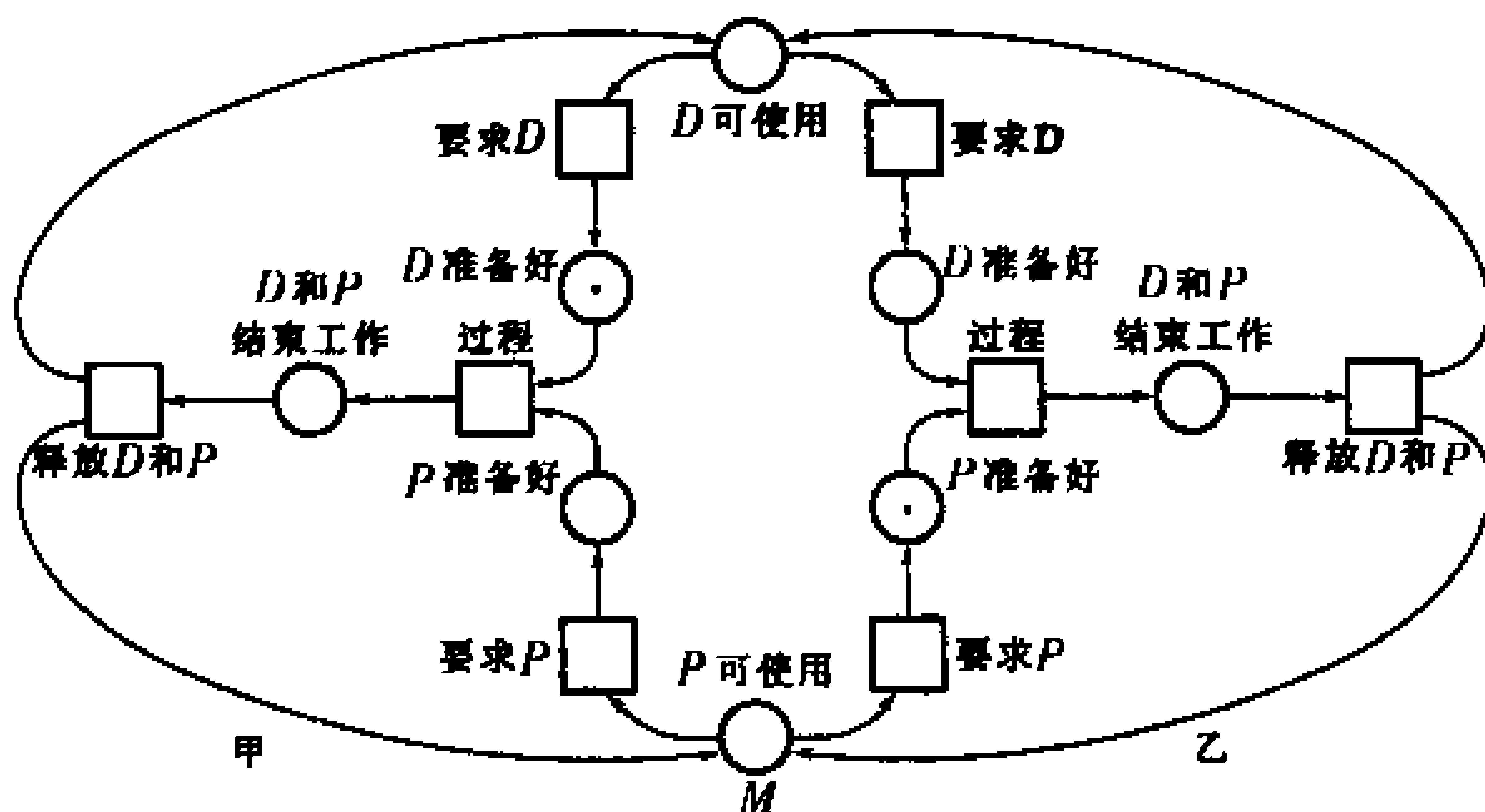


图 6-21

任何转移序列，P 永远不会死锁. 实际上，可以通过某些附加的启动序列的处理启动任何转移.

例如，图 6-18 中标记 M 是活动的. 因为对标记 M 能够启动的唯一转移是 t_1，它产生标记 M'. 对标记 M' 能够被启动的唯一转移是 t_2，它产生标记 M''，对标记 M'' 能够被启动的唯一转移是 t_3，它产生标记 M，显然从标记 M 开始，任意启动序列产生标记 M、M' 和 M'' 之一，并且可以通过对图 6-18 的处理，启动任意转移 t_1，t_2 或 t_3. 所以如图 6-18 所示的标记 M 是活动的.

定义 6 设 Petri 网标记为 M，如果存在某个正整数 n，使得在任何启动序列中，都没有位置能获得多于 n 个记号，则称标记 M 对于该 Petri 网是**有界的**. 如果标记 M 是有界的，并且在任意启动序列中，没有位置能得到多于一个的记号，则称标记 M 是**安全的**.

假如每个位置表示一个寄存器存储一个计算机字，又假如初始标记是安全的，则可以保证寄存器存储容量一定不会超过它所能表示的范围. 即寄存器的存储量是不会越界的.

例如，如图 6-18 所示的标记是安全的．如图 6-22 所示的标记 M 不是安全的．因为在图 6-22 中，如果转移 t_1 被启动，则位置 P_2 有两个记号．

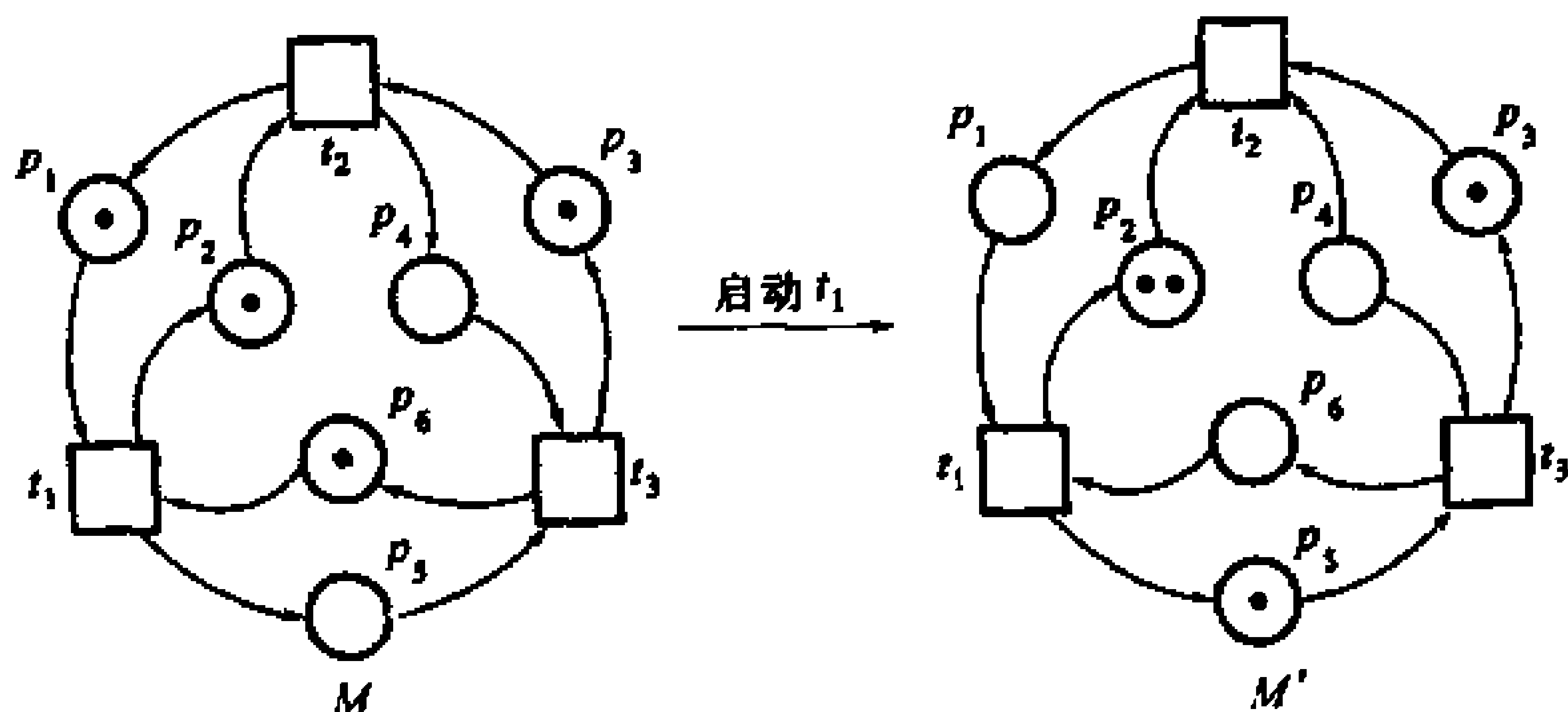


图 6-22

习　题　六

1．填入缺写的边流量，使结果是给定网络的流，并确定流的值．

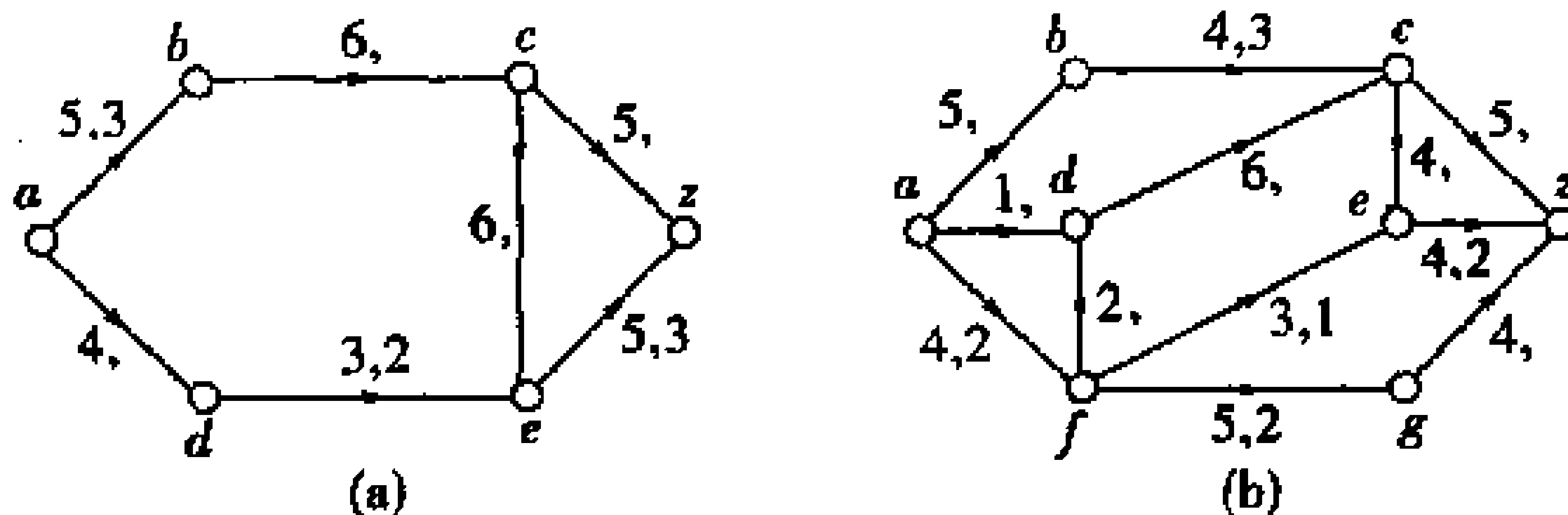


图 6-23

2．图 6-24 表示一个输油网络，其中从三个油井 w_1、w_2 和 w_3 向三个炼油厂 A，B 和 C 输送原油．在边上表明中间系统的容

量.顶点 b,c,d,e 和 f 表示中间泵站.将这个系统刻划为网络模型.若 w_1 最多输送 2 个单位,w_2 最多输送 4 个单位,而 w_3 最多输送 7 个单位,而城市 A 需求 4 个单位,城市 B 需求 3 个单位,城市 C 需求 4 个单位,且中间泵站最多能够输送 6 个单位,其网络模型又如何刻画?

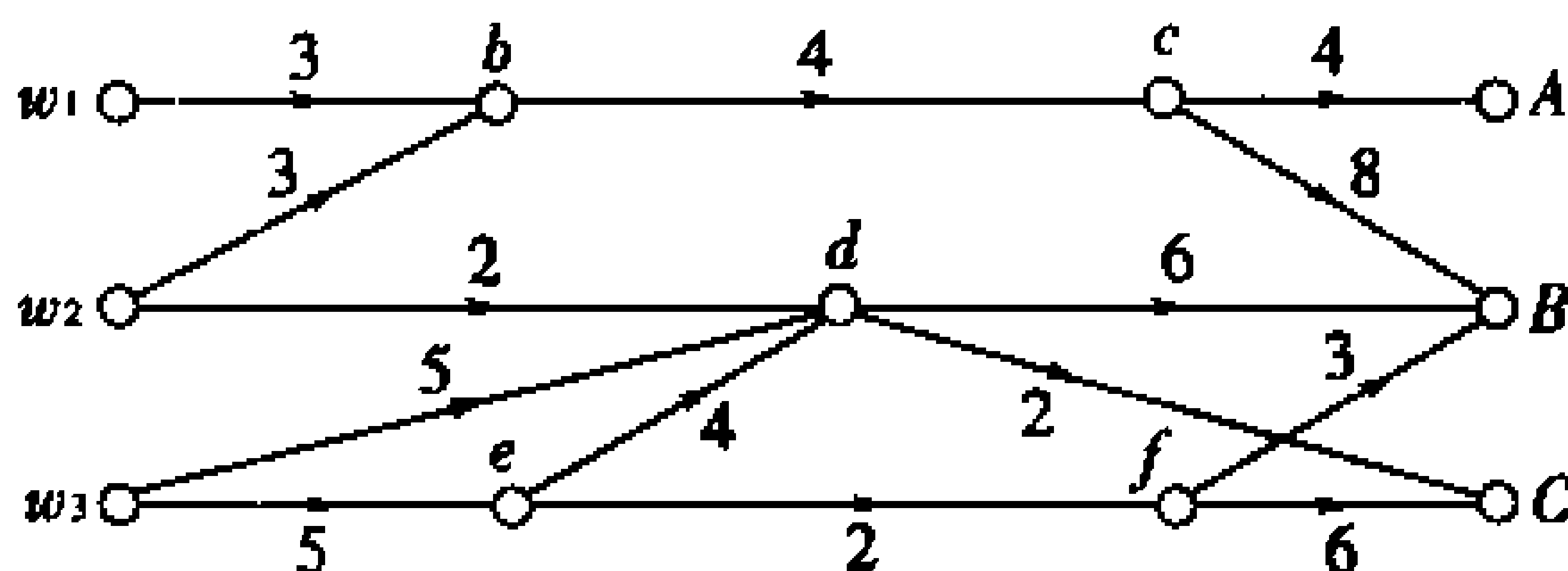


图 6-24

3. 从城市 A 到城市 B 存在两条路,一条经过城市 B,而另一条经过城市 C.在上午 7:00 到 8:00 期间,平均旅行时间是

A 到 B	(30 分钟)
A 到 C	(15 分钟)
B 到 D	(15 分钟)
C 到 D	(15 分钟)

道路上的最大容量是

A 到 B	(1000 辆)
A 到 C	(3000 辆)
B 到 D	(4000 辆)
C 到 D	(2000 辆)

将城市 A 到 D 在早上 7:00 到 8:00 期间的交通流表示为网络.

4. 给出一个网络例子,它恰具有两个最大流,其中每个 F_{ij} 是一个非负整数.

5. 求出图 6.1 所示切割$(p,\bar{p})$的容量,并确定该切割是否最

小.其中(a),$p=\{a,d\}$;(b)中,$p=\{a,b,c,d\}$.

6. C_{ij}表示网络G的非负整型容量,m_{ij}表示G的非负整型最小边流量需求,流F对于所有边$\langle i,j\rangle$必须满足$m_{ij}\le F_{ij}\le C_{ij}$.

试给出一个网络的例子,使对所有的边$\langle i,j\rangle$都有$m_{ij}\le C_{ij}$,都不存在流量.

7. 定义 $c(\bar{p},p)=\sum\limits_{i\in \bar{P}}\sum\limits_{j\in P}c_{ij}$, $m(p,\bar{p})=\sum\limits_{i\in P}\sum\limits_{j\in \bar{P}}m_{ij}$, $m(\bar{p},p)=\sum\limits_{i\in \bar{P}}\sum\limits_{j\in P}m_{ij}$,其中$c_{ij}$与$m_{ij}$为上题所述.

(1)证明对任意流F,任意切割$(p,\bar{p})$满足

$$m(p,\bar{p})-c(\bar{p},p)\le val(F)\le c(p,\bar{p})-m(\bar{p},p)$$

(2)如果G中存在一个流,则G中存在最大流,且其值为$\min\{c(p,\bar{p})-m(\bar{p},p)\mid(p,\bar{p})\text{是}G\text{的切割}\}$.

8. 假设G有一个流F,设计一个求出G中最大流的算法.

9. 如果F是网络G的流,$(p,\bar{p})$是G的切割,且$(p,\bar{p})$的容量超过F的值,则切割$(p,\bar{p})$不是最小且流F不是最大.如果正确,给出证明;否则,给出反例.

10. 已知一个网络中从源a到汇z的一条路,如图6-25所示,通过修改该路中边上的流量,求出最大可能的流量增量.

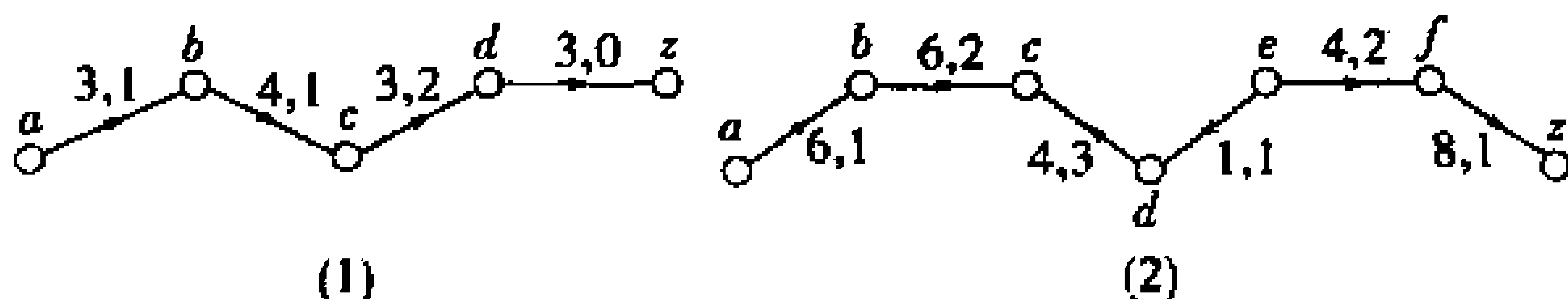


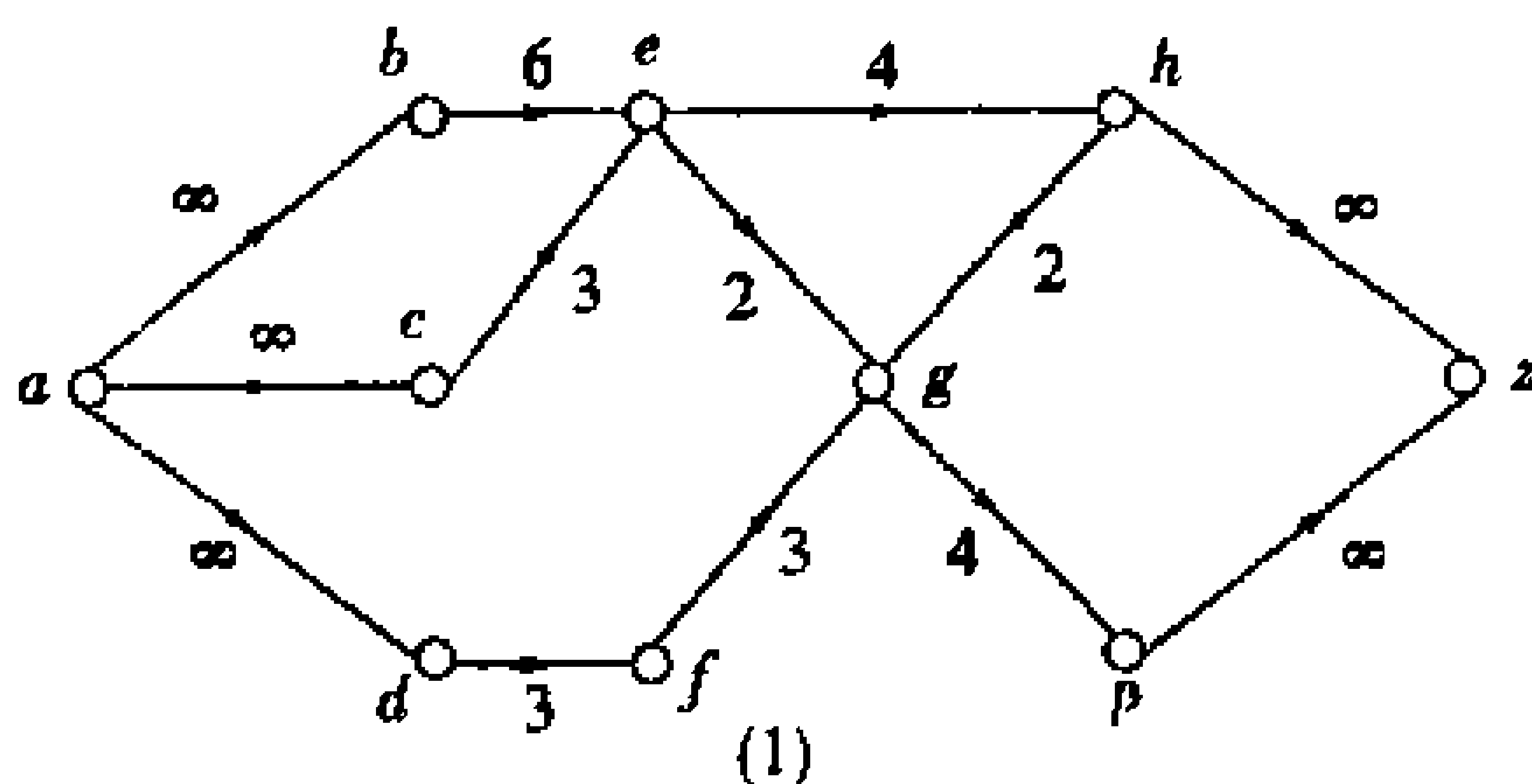
图6-25

11. 求出图6-26中每个网络的最大流.

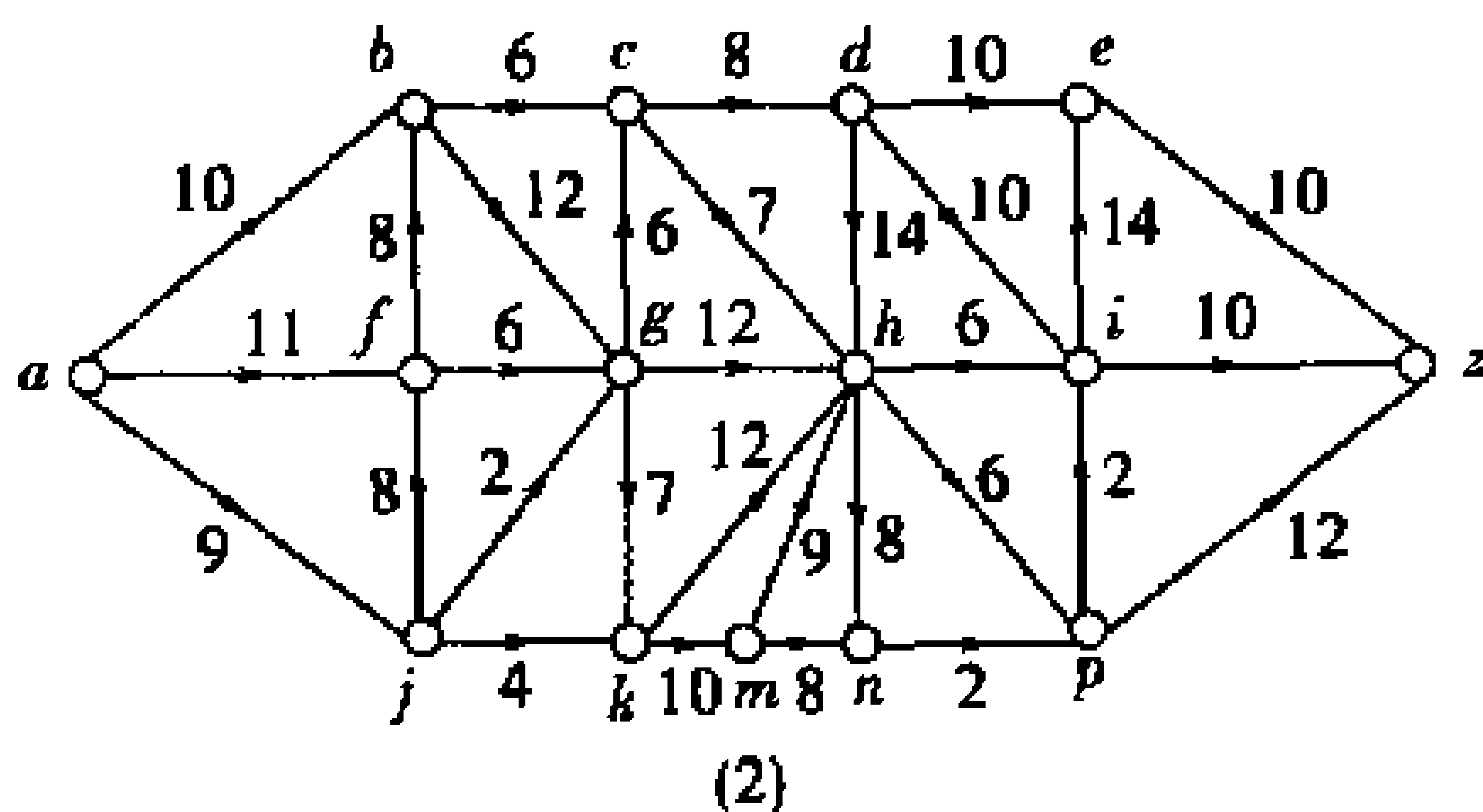
12. 从已知流开始求出图6-27中每个网络的最大流.

13. 设G是整容量网络.证明

(1)在标号法中,若每次修正流F'是通过最短F可增路获得

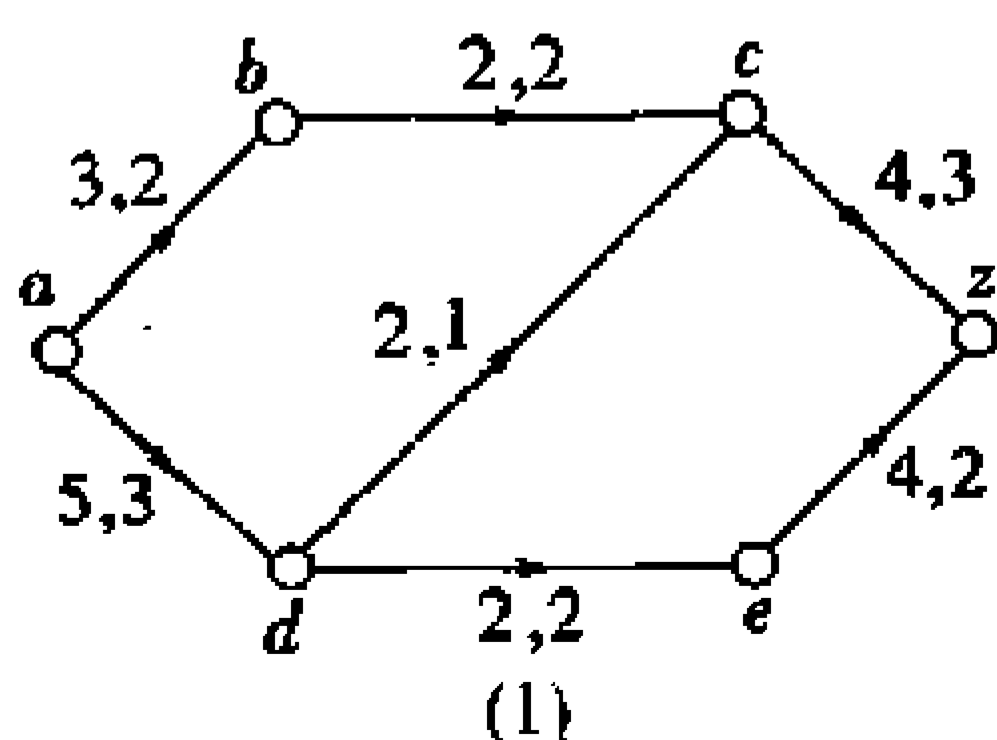


(1)

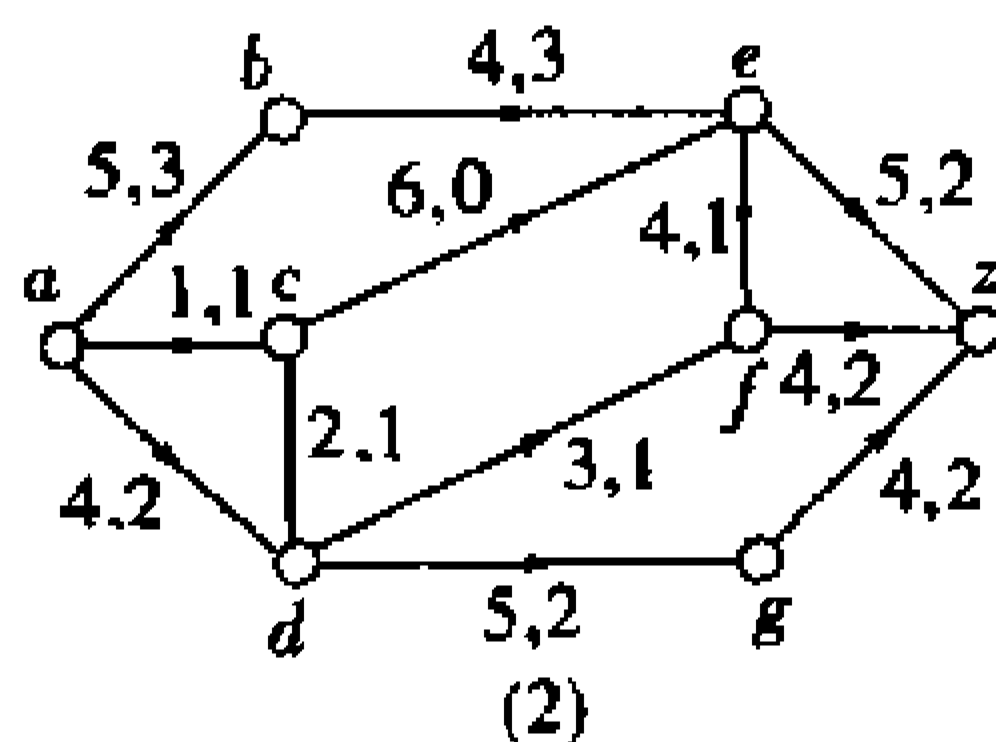


(2)

图 6-26



(1)



(2)

图 6-27

的,则标号程序最多进步$\frac{1}{2}|E(G)|\cdot|V(G)|$次就可以获得最大流.

(2)标号法是有效算法,其复杂度为$O(|E(G)|\cdot|V(G)|^2)$.

14. 设 G 是直径为 2 的简单无何图,$x,y\in V(G)$,证明:G

中至少存在 $\min\{d_G(x),d_G(y)\}$ 条边不交 xy 路，并且每条路的长度不超过 4.

15. 已知 G 是有向图，举例说明下述论述不真：

(1)若 D 中任何 xy 有何路和 yx 有向路都有公共边，则存在边 $e\in E(G)$，使每条 xy 有向路和 yx 有向路都含边 e.

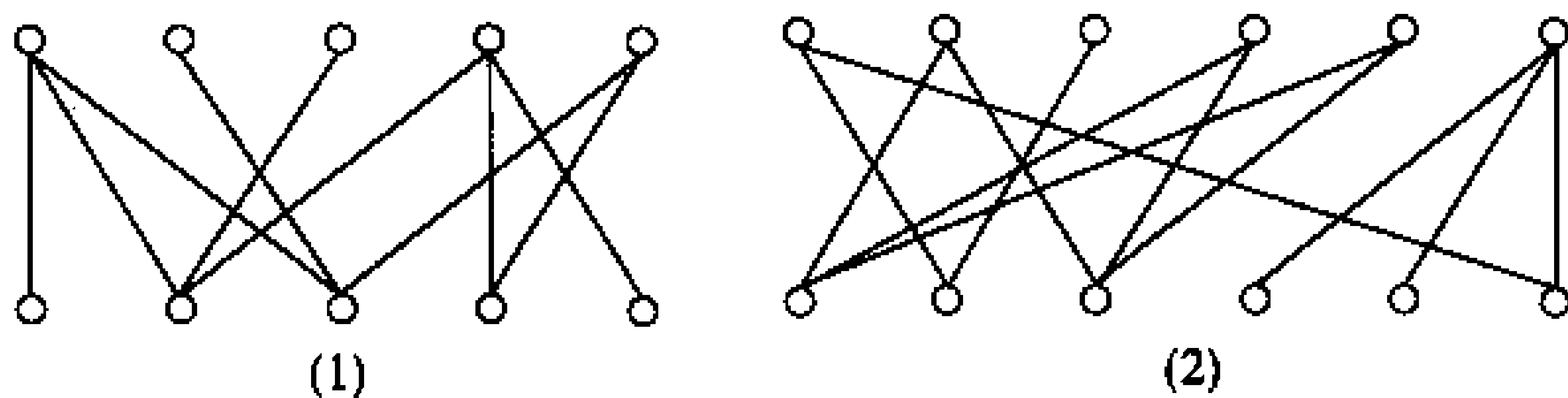
(2)若 G 中有 $k(k\geqslant 1)$ 条边不交的 xy 有向路，则 G 中存在 k 条边不交的 yx 有向路.

(3)若 $\eta_G(x,y)\geqslant k(k\geqslant 1)$ 且 $\eta_G(y,x)\geqslant k$，则 G 中存在 $2k$ 条边不交的有向路 $p_1,p_2,\cdots,p_k,Q_1,Q_2,\cdots,Q_k$ 使 p_i 为 xy 有向路且 Q_i 为 yx 有向路 $(i=1,2,\cdots,k)$

16. 若有向图 G 中异于 x 和 y 的顶点都是平衡点(即入度等于出度的点)，且 $d_G^-(x)-d_G^+(x)=k$，则 G 中有 k 条边不交 xy 有向路.

17. 若 G 是有向平衡图(每个顶点都是平衡点的有向图)，则 15 题中三个论述都成立.

18. 判定下列两个二分图是否有完美匹配. 若没有完美匹配，求出它的一个包含所有最大度点的最大匹配.



(1) (2)

图 6-28

19. 应聘者 A 胜任工作 y_1,y_2,y_4 和 y_5；B 胜任 y_1,y_4 和 y_5；C 胜任工作 y_1,y_4 和 y_5；D 胜任工作 y_1 和 y_5；E 胜任工作 y_2,y_3 和 y_5；F 胜任工作 y_4 和 y_5. 对于此情况回答下列问题：

(1)将它用匹配网络模型来描述.

(2)求出最大匹配.

(3)存在完美匹配吗?

20. 如图 6-29 所示网络中,每条边旁的数是容量和费用,求其最小费用最大流.

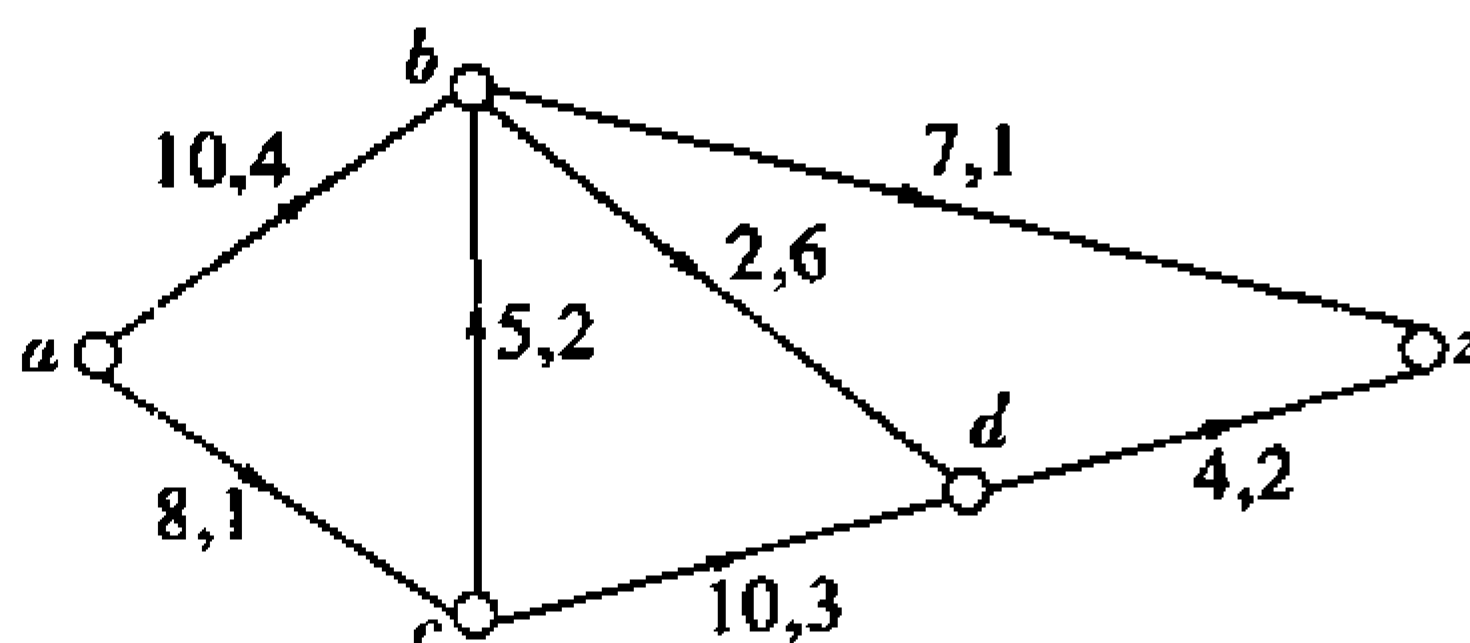


图 6-29

21. 设有三台机器 m_1, m_2 和 m_3,有两个操作人员 A 和 B, A 能操作 m_1 和 m_2, B 能操作 m_1 和 m_3,加工顺序:第一阶段在 m_1 上加工,第二阶段在 m_2 或 m_3 上加工,加工开始必须具备(1)有任务在等待;(2)机器空闲. 试构造该问题的 Petri 网.

22. 某问题的算法如下:

s_1: $y_1 \leftarrow a$, $y_2 \leftarrow b$, $y_3 \leftarrow 1$;

s_2:若 $y_1 > 0$ 转 s_3,否则转 s_5;

s_3:若 y_1 是偶数则转 s_4,否则作:

$$y_3 \leftarrow y_3 * y_2; y_1 \leftarrow y_1 - 1;$$

s_4: $y_2 \leftarrow y_2^2$, $y_1 \leftarrow y_1 - 2$;转 s_2;

s_5:输出 y_3,结束.

23.5 个哲学家围着圆桌坐下,每两个哲学家之间都放着一双筷子. 吃中国菜要一双筷子,所以每位哲学家要吃中国菜必须取他左右的一双筷子,如果每位都取他左边一根等待另一根,则将陷入死锁. 用 Perti 网叙述问题的解.

24. 用 Petri 网为每个程序建立模型. 提供一个标记代表程序执行之前的情形.

(1) $A=1$　　　　(2) $A=2$

$B=2$　　　　$B=A+A$

$C=A+B$　　　　$C=3$

　　　　$D=A+A$

$C=C+1$　　　　$C=A+B+C$

25. 对图 6-30 中每个标记的 Petri 网回答下列问题.

(1)哪一个转移是使能的?

(2)求出由激发 t_1 产生的标记.

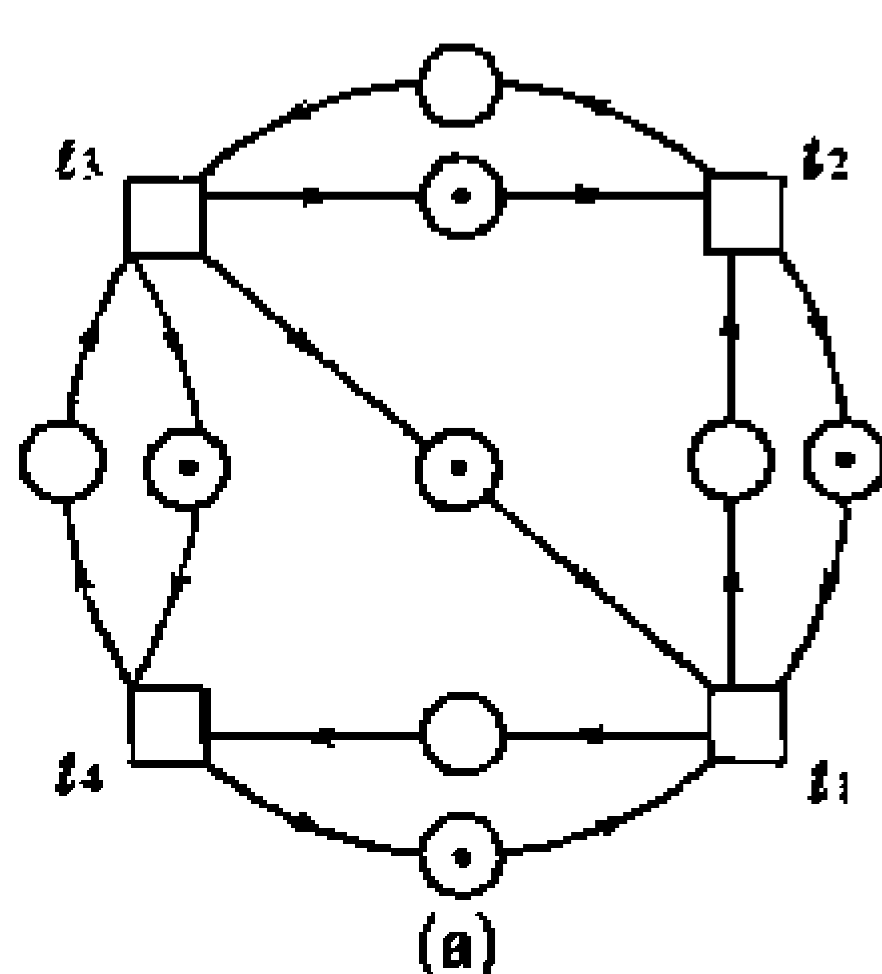
(3) M 是活动的吗?

(4) M 安全吗?

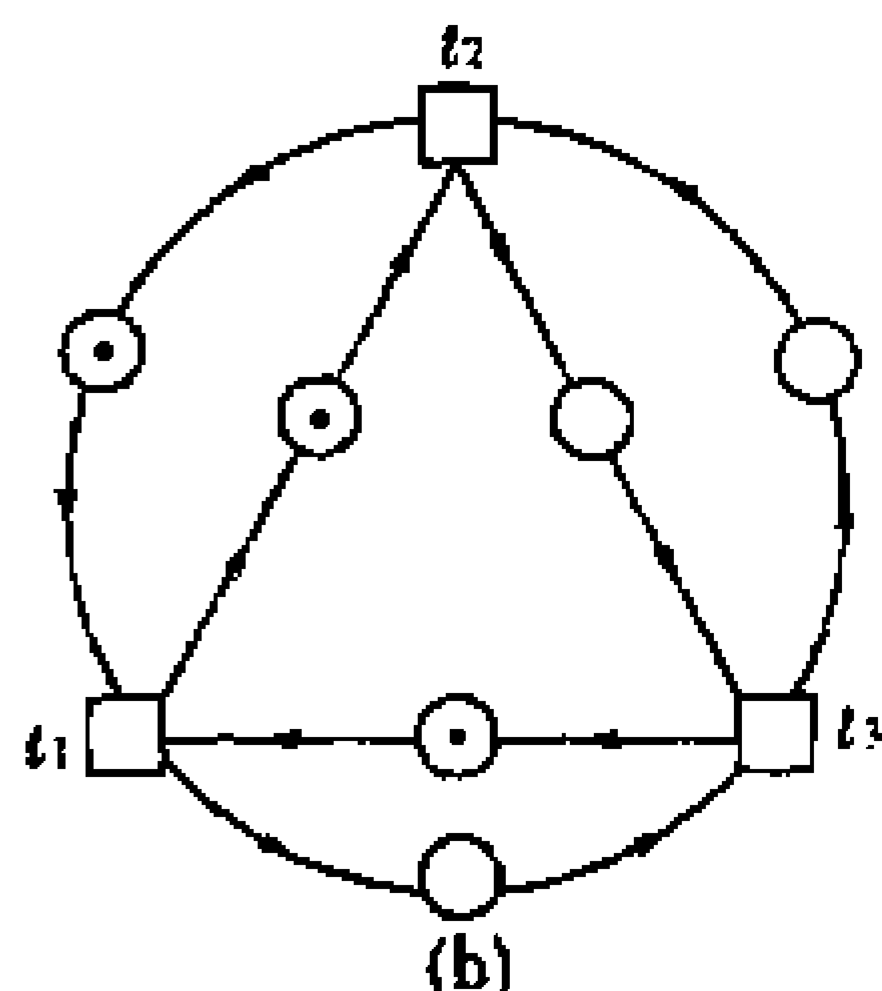
(5) M 有界吗?

(6)描述所有标记从 M 可达.

(7)显示一个标记(并非在每个位置放置零个记号的标记)不是从 M 可达的.



(a)



(b)

图 6-30

26. 给出一个 Petri 网的实例,它具有的一个安全的但不是活动的标记.

27. 给出一个 Petri 网的实例,它具有的一个有界的但不是安全的标记.

附录1 符号集

符号	含义
$A \cap B$	集合A和B的交集
$A \cup B$	集合A和B的并集
$A - B$	集合B相对于A的补集
$\bar{A}$	集合A的绝对补
$A \oplus B$	集合A和B的对称差
$A \times B$	集合A和B的笛卡尔积
$\|A\|$	集合A的基数
$\langle a_1, a_2 \rangle$	二元组(序偶)
$\langle a_1, a_2, a_3 \rangle$	三元组
$\langle a_1, a_2, \cdots, a_n \rangle$	n元组
$G_1 \cup G_2$	图G_1与G_2的并
$G_1 \cap G_2$	图G_1与G_2的交
$G_1 - G_2$	图G_1与G_2的差
$G_1 \cdot G_2$	图G_1与G_2的乘积
$G_1 \oplus G_2$	图G_1与G_2的环和
$G_1 \times G_2$	图G_1与G_2的笛卡尔积
$G_1 \subseteq G$	图G_1与G_2的子图
$G[V_1]$	V_1的导出子图
$G[E_1]$	E_1的导出子图
$C(G)$	图G的闭包
$G - V_1$	从G中删去V_1中点及这些点关联的边,所得的子图
$G - E_1$	从G中删去E_1中边所得子图

$G-x$	$G-\{x\}$，x 为 G 的结点
$G-e$	$G-\{e\}$，e 为 G 的结点
$G\cdot xy$	收缩图 G 的不邻接顶点 x 与 y 为一新顶点，且与 x，y 关联的边均与该新顶点关联.
$G:e$	收缩图 G 的边 e
$\bar{G}$	G 的补图
$G_1 \cong G_2$	图 G_1 与 G_2 同构
K_n	n 个结点的完全图
$K_{m,n}$	完全二分图
$K_{n_1,n_2,\cdots,n_m}$	完全 m 分图
K_n^*	n 个结点的完全有向图
Q_n	n(维)立方体
C_n	n 个结点的圈图
W_n	n 个结点的轮图
$\deg(x)$ 或 $d(x)$	结点 x 的度数
$\deg^+(x)$ 或 $d^+(x)$	结点 x 的入度
$\deg^-(x)$ 或 $d^-(x)$	结点 x 的出度
$\Delta(G)$	G 的最大顶点度
$\delta(G)$	G 的最小顶点度
$\Delta^+(G)$	G 的最大顶点入度
$\delta^+(G)$	G 的最小顶点入度
$\Delta^-(G)$	G 的最大顶点出度
$\delta^-(G)$	G 的最小顶点出度
$N_S(x)$	结点 x 在 S 中的邻域
$N_G(s)$	S 的邻集
$[V_1, V_2]$	图 G 的两个端点分别在顶点集 V_1 和 V_2 中的边的集合
$d(x,y)$	图 G 中结点 x 到 y 的距离
D	图 G 的直径
$K(G)$	G 的(点)连通度

$\lambda(G)$	G 的边连通度
$W(G)$	图 G 的连通分支数
$\vec{W}(G)$	图 G 的强连通分支数
$O(G)$	G 的奇阶连通分支数
$B(k,n)$	德布鲁英图(序列)
$W(T)$	二叉树 T 的权
$V_G(H_1,H_2)$	H_1 和 H_2 在 G 中的接触点集
$\chi(G)$	图 G 的色数
$\chi'(G)$	图 G 的边色数
$f(G,k)$	图 G 的色数多项式
$\theta(G)$	简单图 G 的厚度
$\alpha(G)$	图 G 的独立数
$\alpha'(G)$	图 G 的匹配数
$\beta(G)$	图 G 的点覆盖数
$\beta'(G)$	图 G 的边覆盖数
$\gamma(G)$	图 G 的支配数
$\zeta_G(x,y)$	图 G 中内部点不交的 xy 路的最大条数
$\eta_G(x,y)$	图 G 中边不交的 xy 路的最大条数
$\lambda_G(x,y)$	图 G 中最小 xy 截边集的边数
$K_G(x,y)$	图 G 中最小 xy 分离集的顶点数目
$[x]$	不超过 x 的最大整数
$\lceil x \rceil$	不小于 x 的最小整数

附录 2　习题解答

习　题　一

1　**证明**　建立图 $G=\langle V,E\rangle$，其中 $V=S=\{x_1,x_2,\cdots,x_n\}$，$V$ 中的两个顶点有一边连接当且仅当这两个点之间的距离恰好为 1。则所得图 G 是一个简单图，S 中距离为 1 的点对数就是 G 的边数。因此只需证明 $|E(G)|\leq 3n$。

任取 $x_i\in V$，G 中与 x_i 相邻的顶点为 $x_{i_1},x_{i_2},\cdots,x_{i_k}$（$k=d(x_i)$）。则 $x_{i_1},x_{i_2},\cdots,x_{i_k}$ 分布在以 x_i 为圆心的单位圆周上。所以 $k\leq 6$，即 $d(x_i)\leq 6$，$i=1,2,\cdots,n$。

而 $2|E(G)|=\sum_{i=1}^{n}d(x_i)\leq 6n$，故 $|E(G)|\leq 3n$。

2　**证明**　作图 $G=\langle V,E\rangle$，其中 V 中有 n 个顶点，分别代表参加会议的 n 名代表。V 中两个顶点相邻当且仅当这两个顶点所对应的代表互相认识。则只要证明 G 是一个正则图。

3　**证明**　因为每个碳氢化合物的分子由氢与碳原子组成，其原子价分别为 1 阶和 4 阶。让每个原子对应于图的一个顶点，如果两个原子是连接着的，那么对应的两个顶点就相邻。

4　**解**　建立一个有向图 $G=\langle V,E\rangle$，其中 $V=\{x_1,x_2,\cdots,x_n,y_1,y_2,y_n\}$，如果 x_i 与 y_j 进行过一场比赛，就在 x_i 与 y_j 之间连一条有向边，其方向从胜者指向负者。则 G 中的每一条边对应一场比赛，G 中边的数目就是这次比赛的次数。根据比赛规则，

每一名选手最多输一场，所以 G 中每个顶点的入度至多为 1，但 x_n 与 y_n 必有一个入度为 1，另一为 0。因而我们有 $|E| = \sum_{i=1}^{n}(d^-(x_i)+d^-(y_i)) \leq 2n-1$。即至多进行 $2n-1$ 场比赛就可以确定胜负。

5 解 我们用 n 个顶点来代表 n 个药箱，两个顶点之间连接的边数等于这两个顶点所对应的两个药箱所含有相同化学品的种数。按题意，所得图 G 是一个 n 阶完全图，G 中每一条边代表一种化学品。因此在这个化学实验室中，每个药箱装有 $n-1$ 种化学品。整个实验室共有 C_n^2 种不同的化学品。

6 证明 作一个二分图 $G=\langle A\cup B,E\rangle$，$A$ 中的 n 个顶点代表 A 国的 n 个留学生，B 中的 n 个顶点代表 B 国的 n 个留学生。如果一个 A 国的学生与一个 B 国的学生一起跳过舞，就在相应的两个点之间连一条边。所得二分图 G 是一个简单图，且 $|A| = |B| = n > 2$，对于 A 中每个点 x，$1 \leqslant d(x) < n$，对 B 中的每个点 y，$d(y) \geqslant 1$。

在 A 中取一个度数最大的点，在 B 中取一个与 x 不相邻的点 y'，由于 $d(y') \geqslant 1$，令 x' 是 A 中与 y' 相邻的顶点，则 $d(x') \leqslant d(x)$，$x' \neq x$，即有 $|N_G(x')| \leqslant |N_G(x)|$，又 $y' \in N_G(x') - N_G(x)$，所以 $N_G(x) - N_G(x') \neq \varnothing$。取 $y \in N_G(x) - N_G(x')$，则 y 与 x 相邻而与 x' 不相邻。这四个点 x、x'、y、y' 所对应的 4 个留学生即为所求。

7 解 用 F 代表摆渡人，W 表示狼，S 表示羊，H 表示干草。于是，根据题意，集合 $\{F,S,W,H\}$ 中，允许留在原岸边的子集是 $\{F,S,W,H\}$，$\{F,S,W\}$，$\{F,S,H\}$，$\{F,W,H\}$，$\{F,S\}$，$\{W,H\}$，$\{S\}$，$\{W\}$，$\{H\}$，$\varnothing$。我们可以构造一个图 G（如图 1 所示），其中 G 以这些子集为顶点，摆渡前在原岸边的子集与经过一次摆渡后仍在原岸边的子集间连上一条边。

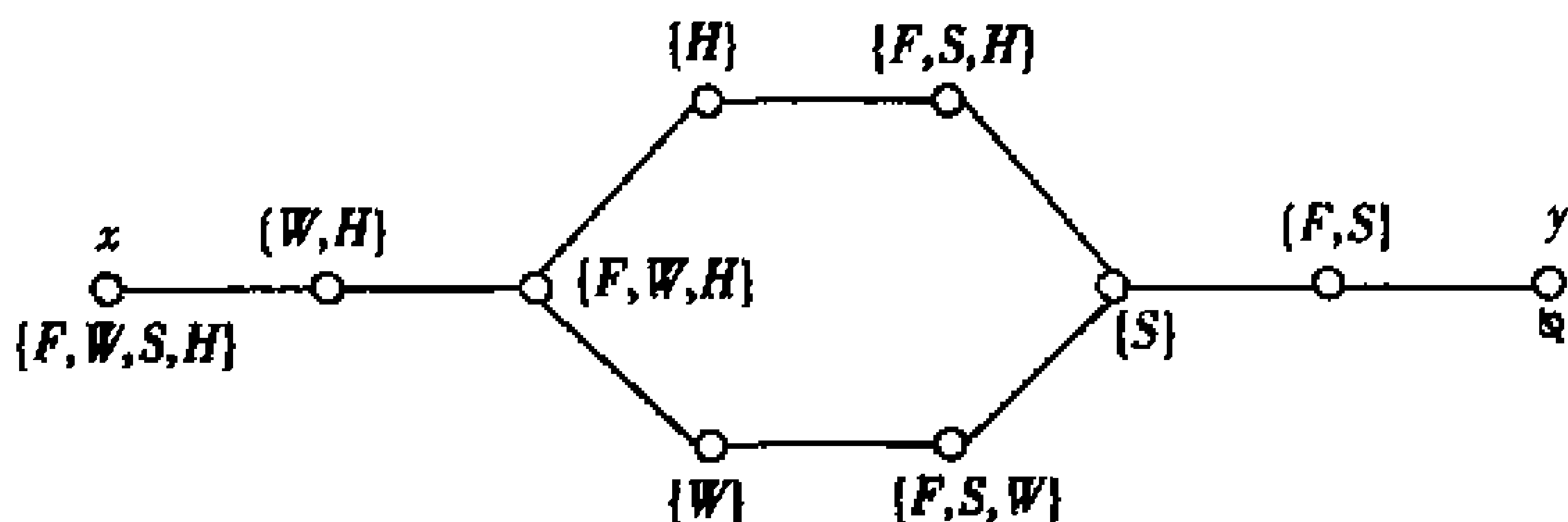


图 1

问题的解就是要我们找出图 G 中一条从顶点 $x=\{F,W,S,H\}$ 到顶点 $y=\emptyset$ 的路。

8　**证明**　把每个方格作为顶点并标上该方格所填的数，相邻的方格对应相邻的顶点，所得图记为 G。只要证明该图 G 中存在一条边 e，而 e 的两个端点的标号数之差的绝对值不小于 $\lceil \frac{n+1}{2} \rceil$。

容易算得图 G 的直径为 $D=(n-1)+(n-1)=2(n-1)$，即 G 中任意两个顶点之间有一条长度不超过 $2n-2$ 的路连接它们。

设 G 中任意一条边的两端点标号数之差的绝对值不小于 $\lceil \frac{n+1}{2} \rceil$，则 G 的任意两点的标号数之差的绝对值不超过 $2(n-1)(\lceil \frac{n+1}{2} \rceil -1)<2(n-1)(\frac{n+1}{2}+1-1)=n^2-1$

但 G 中必有一个顶点的标号数为 n^2，一个顶点的标号数是 1。这两个顶点的标号数之差的绝对值是 n^2-1。矛盾。

9　**证明**　设这 $(n+1)^2$ 项递增的自然数列为 $v_1,v_2,\cdots,v_{(n+1)^2}$。作一有向图 $D=\langle V,E\rangle$，其中 $V=\{v_1,v_2,\cdots,v_{(n+1)^2}\}$，若 v_i 整除 v_j，就在 D 中引一条从 v_i 到 v_j 的边 $\langle v_i,v_j\rangle(i\neq j)$。显然，图 D 无有向圈，而且具有结论(1)的子列对应 D 中的一条

长为 $n+2$ 的有向路；具有结论(2)的子列对应 D 中的 $n+1$ 个互不相邻的顶点。

对每一个顶点 v_i，考虑以 v_i 为起点的所有有向路，记其中最长的一条有向路的长为 $l(v_i)$。如果有某个 $l(v_i)\geqslant n+2$，则结论成立。如对一切 v_i，$l(v_i)\leqslant n+1$，记满足 $l(v_i)=j(0\leqslant j\leqslant n+1)$ 的顶点 v_i 的个数为 $a(j)$。则

$$a(0)+a(1)+\cdots+a(n+1)=|V|=(n+1)^2$$

而 $\frac{(n+1)^2}{n+2}=\frac{n(n+2)+1}{n+2}=n+\frac{1}{n+2}>n$

所以必有一个 j_0，$0\leqslant j_0\leqslant n+1$，使 $a(j_0)\geqslant n+1$。即在 D 中至少存在 $n+1$ 个顶点 $v_{j_1},v_{j_2},\cdots,v_{j_{n+1}}$，使 $l(v_{j_k})=j_0$，$k=1,2,\cdots,n+1$。由假设，D 中以 v_{j_k} 为起点的最长有向路的长为 j_0，现可断定这 $n+1$ 个顶点互不相邻。否则，如有 $\langle v_{j_k},v_{j_i}\rangle\in E$，$(0\leqslant k\neq i\leqslant n+1)$，则

$$l(v_{j_k})\geqslant l(v_{j_i})+1=j_0+1 \qquad 矛盾。$$

10 证明 构造图 $G=\langle V,E\rangle$ 如下：V 就取平面中给定的 $2n$ 个点，两个不同的顶点如果含在同一个圆面上，就在这两个顶点之间连上一条边(边也含在这个圆面上)。所得图 G 是一个简单图，而且每个顶点的度至少是 n，即

$\delta(G)\geqslant n=[\frac{2n}{2}]$，从而 G 是连通图，所以 G 中任两点之间有一条路连接。

11(2) 解 无向 4 阶自补图和无向 5 阶自补图如图 2 所示。

13 证明 (1) $\sum_{i=1}^{n} d(v_i)=3n=2m$，故 $n=\frac{2}{3}m$。又 $m=3n-6=3\times\frac{2}{3}m-6=2m-6$，所以 $m=6$，从而 $n=4$。而 $m=6$，

$n=4$ 的无向简单图即同构于无向完全图 K_4。

(2) $\sum_{i=1}^{n} d(v_i)=3\times6=18=2m$,故 $m=9$

此时 $n=6,m=9$,且每个结点的度数为 3,满足此条件的无向简单图如图 3(a)与(b)所示,显然它们不同构。

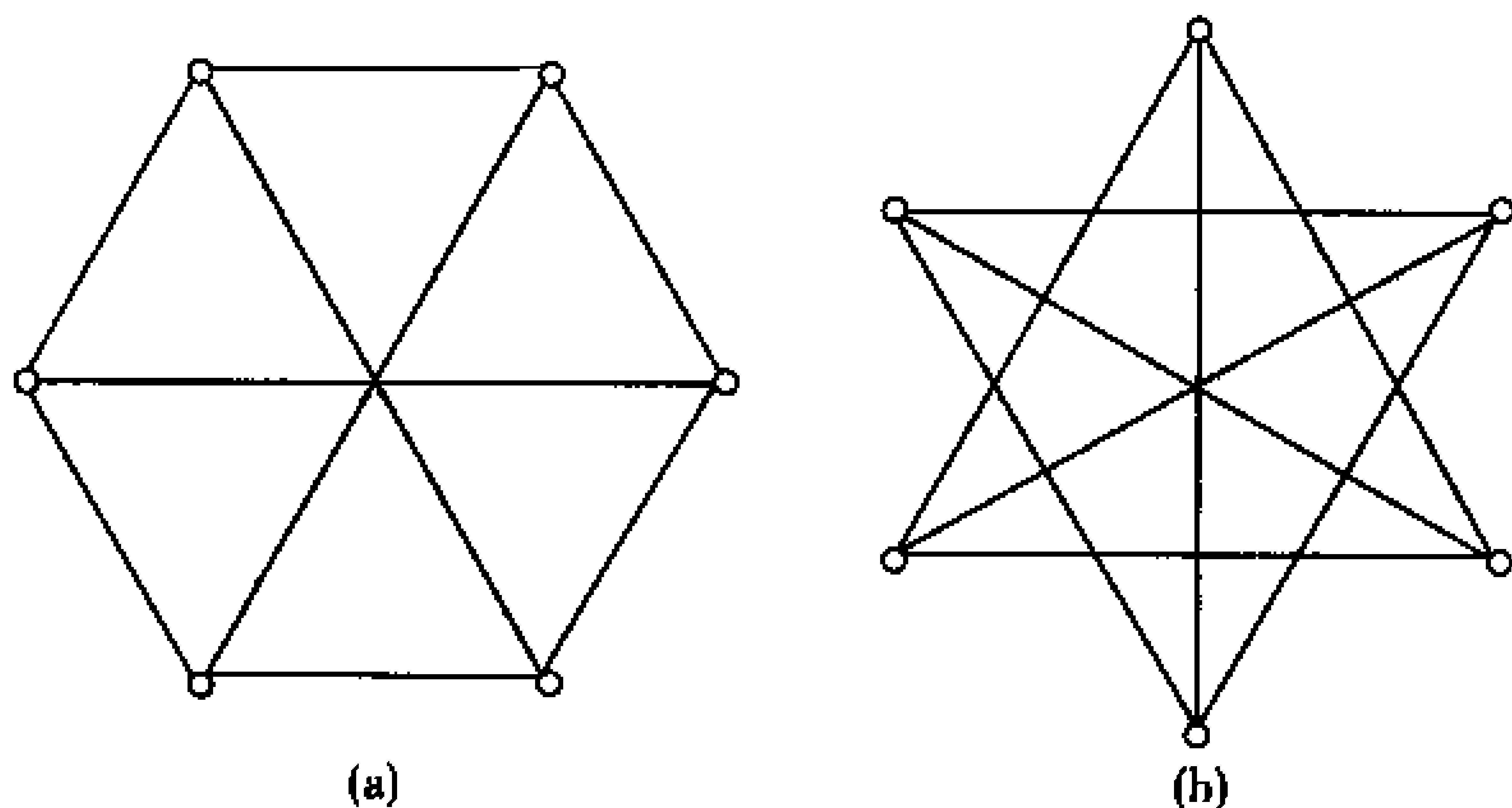


图 3

15 证明 设不连通的无向图 $G=\langle V,E\rangle$ 仅有两个连通分支,这两个连通分支的结点集分别为 V_1 和 V_2,且 $V_1=\{u_1,u_2,\cdots,u_r\}$, $V_2=\{v_1,v_2,\cdots,v_s\}$

(1)设 $u_i\in V_1,v_j\in V_2$,显然边 $\{u_i,v_j\}\notin E$,从而边 $\{u_i,u_j\}$

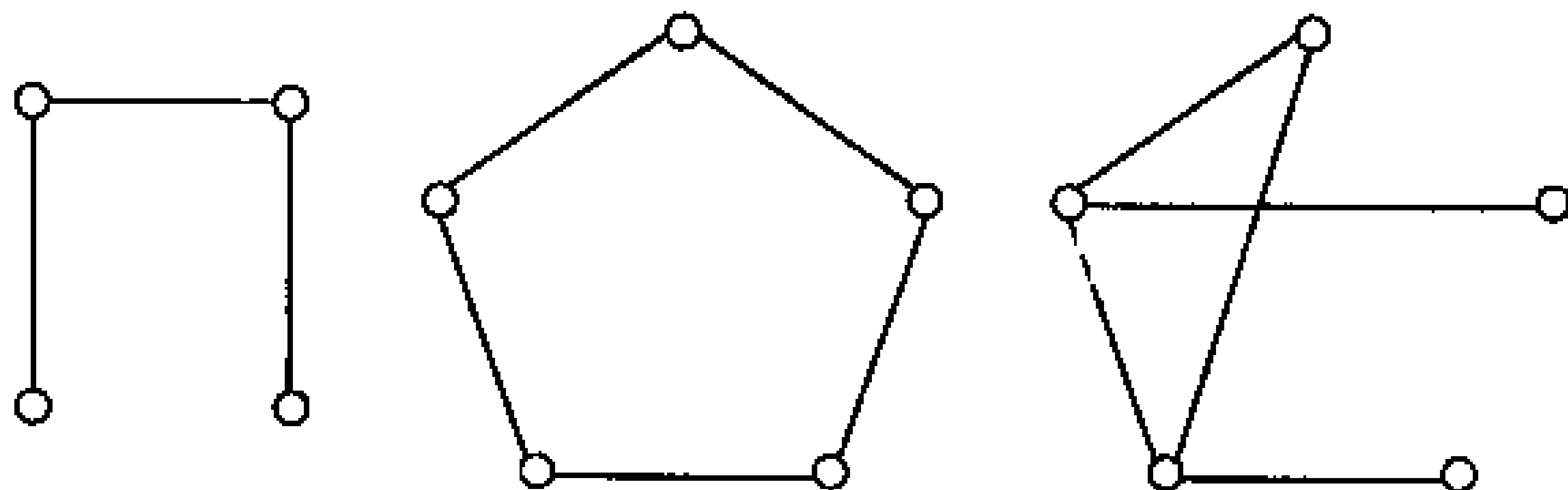


图 2

$\in E(\bar{G})$。

(2)设 $u_i, u_j \in V_1$ 或 $v_i, v_j \in V_2$,不妨设 $u_i, u_j \in V_1$, $\forall v_k \in V_2$,显然边 $\{u_i, v_k\} \notin E$, $\{u_j, v_k\} \notin E$,从而边 $\{u_i, v_k\} \in E(\bar{G})$, $\{u_j, v_k\} \in E(\bar{G})$,因此 u_i 与 u_j 可通过无向路 u_i, v_k, u_j 相连通。

16 **解** 一定相等。因 $n>2$ 为奇数,则对于奇数个结点的 n 阶无向完全图,每个结点的度数必为偶数。若 G 的奇度数结点为 m 个,则对应补图 $\bar{G}$ 在这 m 个结点的度数必为(偶数－奇数)＝奇数。另外,对于 G 中度数为偶数的结点,其在补图 $\bar{G}$ 中,这些结点的度数仍为(偶数－偶数)＝偶数。

20 **证明** 用 9 个顶点 $v_1, v_2, \cdots, v_9$ 表示 9 个数学家,如果 v_i 与 v_j 可以用第 r 种语言对话,则在 v_i 与 v_j 之间连一边并将这条边涂上第 r 种颜色,这样就得到了图 G,它的边涂上了颜色(至多有 27 种)。如果边 $\{v_i, v_j\}$ 和 $\{v_j, v_k\}$ 都涂上了第 r 种颜色,那么边 $\{v_i, v_k\}$ 也涂上了第 r 种颜色。

由已知得每 3 个顶点之间至少存在一条边,每一个顶点 v_i 引出的边至多有三种不同的颜色。只需证明图 G 中至少有一个三角形,这三角形的三条边也是同一种颜色。

用反证法。假设从任一顶点 v_i 引出的边颜色都不相同,因而根据已知条件得 $d(v_i) \leqslant 3(1 \leqslant i \leqslant 9)$。

对于顶点 v_1,由于 $d(v_1) \leqslant 3$,所以至少有 $9-1-3=5$ 个顶点与 v_1 不相邻,不妨设为 v_2, v_3, v_4, v_5, v_6,因 $d(v_2) \leqslant 3$,在 v_3, v_4, v_5, v_6 这 4 个点中必有一个点与 v_2 不相邻,设为 v_3,那么 v_1, v_2, v_3 这三个点之间无边。矛盾。

习 题 二

1 **解** 设 $v_1, v_2, \cdots, v_5$ 分别表示 1～5 年年初更新设备状

态，用 v_6 表示第 5 年年底更新设备状态。从 v_i 到 $v_{i+1}, \cdots, v_6$ 各画一条有向边。边 $\langle v_1, v_2\rangle$ 表示在第 1 年年初购进的设备用到第 2 年年初，权 W_{12} 表示需要的购买费加维修费，即 $W_{12}=11+5=16$。边 $\langle v_1, v_3\rangle$ 表示第 1 年买的设备用到第 3 年年初，权 W_{13} 表示所需总费用，即 $W_{13}=11+5+6=32$ 等等。这样就得到如图 4 所示的图 G，于是设备更新问题转化为求从 v_1 到 v_6 的最短有向路问题。通过计算，$v_1 \to v_3 \to v_6$ 和 $v_1 \to v_4 \to v_6$ 都是最短路，对应的费用为 53 万元，它们都是最优的更新方案。

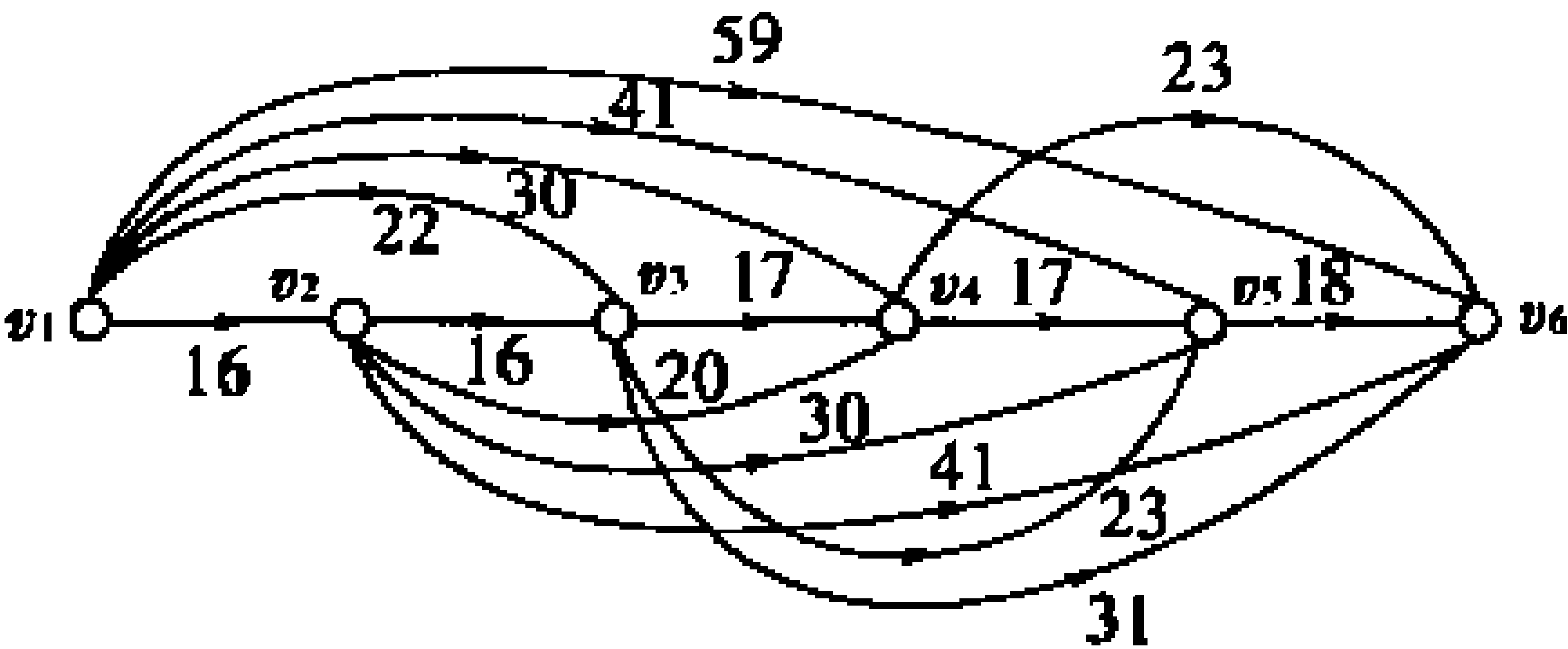


图 4 G

2 **解** 利用 Floyd 算法，首先求出任意两点 v_i, v_j 间的最短路长，如表 1 所示。

表 1

从＼到	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	2	6	7	8	11
v_2	2	0	4	5	6	9
v_3	6	4	0	1	2	5
v_4	7	5	1	0	1	4
v_5	8	6	2	1	0	3
v_6	11	9	5	4	3	0

设想医院建在村庄 v_j，则其他村庄的村民就要分别走 d_{1j}，d_{2j}，…，d_{6j} 的路程，d_{ij} 表示 i 村到 j 村的实际距离，其中必有最大者。对每一点 v_j，求出这个最大值，我们希望建在这些最大值之中的最小值所对应的村庄。这相当于表 1 中每一行元素中的最大值，它们分别是 11,9,6,7,8,11，这些数中 6 最小，即第 3 列的最大元素达到最小，所以医院应建在村庄 v_3，这样其它村庄到该村庄就医距离最多为 6。

设想小学建在 v_j，则其他村庄的小学生所走的总路程就是 $50d_{1j}+40d_{2j}+60d_{3j}+20d_{4j}+70d_{5j}+90d_{6j}$。对每一点，求出这个值，它们的最小值所对应的 v_j 就是所要选择的最佳位置。这相当于将表 1 中每一行元素分别乘上对应村庄里小学生的人数，然后分别求出各列的和，见表 2，其总和最小的列为 v_4，即小学建在村庄 v_4 位置，这样必使所有学生上学所走的总路程最短。

表 2

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	100	300	350	400	550
v_2	80	0	160	200	240	360
v_3	360	240	0	60	120	300
v_4	140	100	20	0	20	80
v_5	560	420	140	70	0	210
v_6	990	810	450	360	270	0
总和	2130	1670	1070	1040	1050	1500

3 **解** 用一条有向边表示一个工序，有向边与有向边的衔接来表示工序和工序的先后关系，用边的权来表示完成这个工序所用的时间，得到一个无回路的有向图 G，如图 5 所示。

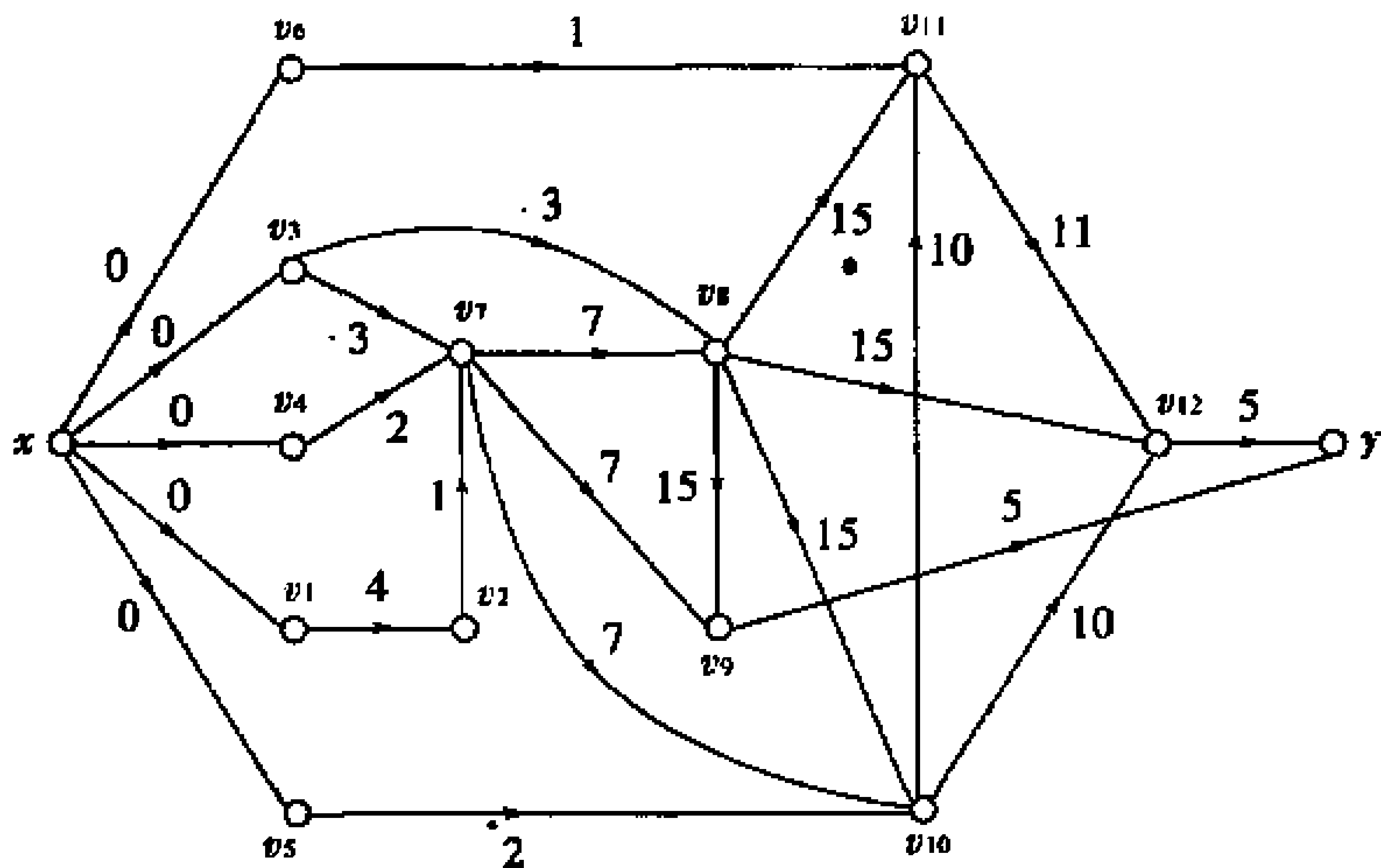


图5 G

x 表示该工程开始的点，y 表示该工程的结束点。显然，图 G 中，从始点 x 到终点 y 的最长有向路上的边所对应工序的完成进度将直接影响到整个工程的进度（这些工序称为关键工序，该有向路称为关键路线）。从始点 x 到各顶点的最长有向路的长度就是以该点为始点的各工序的最早开工时间，也就是说这些工序不能在这个时间之前启动，顶点 v_k 的最早开工时间用 t_k 表示。从每一个顶点到终点 y 也有一条最长有向路，它的长度是从这点开始的工序到项目终止所需的最少时间，用 $\bar{t}_k$ 表示。若用 T 表示关键路线的长，则 $T-\bar{t}_k$ 是从该点开始的工序的最晚时间，晚于这个时间便影响整个工程的预期完成。

对关键工序 v_k，恒有 $t_k=T-\bar{t}_k$，而非关键工序一般有 $t_k<T-\bar{t}_k$。从 t_k 到 $T-\bar{t}_k$ 这一段时间是工序的缓冲时间，在这段时间内开工不影响任务完成。每个工序也有各自的缓冲时间，这对于实际计划安排是有用的。

将图 G 的每条边 e 的权 $W(e)$ 改为 $M-W(e)$，这里 M 是一个充分大的数，得到赋权有向图 G'，这样图 G 中的最长有向路对应于 G' 中相应的最短有向路。

图 G 中，从始点 x 到终点 y 的最长有向路是 $x\to v_1\to v_2\to v_7\to v_8\to v_{10}\to v_{11}\to v_{12}\to y$，它们的长度为 53 周。这些工序是关键工序，其中任一个工序若拖延一周，整个工程的完成就要推迟一周。因此，管理人员应特别注意不让这些工序超过它们的计划期。

从图 G 我们还可以算出各个事项的缓冲时间。比如，考虑工序 v_9（安装生产设备这一工序），由于路 $x\to v_1\to v_2\to v_7\to v_8\to v_9$ 的长度是 27，故工序 v_9 应在 27 周以后才能开工。若完成工序 v_9 用去的时间不是预计的 5 周，而是 26 周，那么路 $x\to v_1\to v_2\to v_7\to v_8\to v_9\to y$ 的长度将是 53，这并不会拖延整个工程的完工时间，因而工序 v_9 的缓冲时间为 $26-5=21$ 周。

4 **解** 用二元组 $\langle b,c\rangle$ 表示桶 B 和桶 C 中水的数量分布。例如 $\langle 0,0\rangle$ 表示 B 和 C 都是空的；$\langle 6,0\rangle$ 表示 B 中有 6 升水，而 C 是空的，等等。由于 b 和 c 都是非负整数且 $b+c\leqslant 12$，所以 $\langle b,c\rangle$ 只有以下 26 种取值：$\langle 0,0\rangle$，$\langle 1,0\rangle$，$\langle 2,0\rangle$，$\langle 3,0\rangle$，$\langle 4,0\rangle$，$\langle 5,0\rangle$，$\langle 6,0\rangle$，$\langle 7,0\rangle$，$\langle 8,0\rangle$，$\langle 9,0\rangle$，$\langle 0,1\rangle$，$\langle 0,2\rangle$，$\langle 0,3\rangle$，$\langle 0,4\rangle$，$\langle 0,5\rangle$，$\langle 1,5\rangle$，$\langle 2,5\rangle$，$\langle 3,5\rangle$，$\langle 4,5\rangle$，$\langle 5,5\rangle$，$\langle 6,5\rangle$，$\langle 7,5\rangle$，$\langle 8,4\rangle$，$\langle 9,1\rangle$，$\langle 9,2\rangle$，$\langle 9,3\rangle$。

用 26 个顶点 x_{bc} 代表这 26 个分布 $\langle b,c\rangle$，顶点 x_{bc} 和 $x_{b'c'}$ 之间有一条边相连 $\Leftrightarrow$ 分布 $\langle b,c\rangle$ 和 $\langle b',c'\rangle$ 之间可以通过一次倾倒而相互得到。例如，对于分布 $\langle 3,5\rangle$，将 c 中 5 升水倒人 B 中，则得分布 $\langle 8,0\rangle$；反之，对于分布 $\langle 8,0\rangle$，用 B 中水注满足 c 便得到分布 $\langle 3,5\rangle$。于是，代表这两个分布的两个顶点 x_{35} 和 x_{80} 之间连一条边。这样得到一个 26 阶简单无向图 G，如图 6 所示。

于是，问题归结为在图 G 中求一条从 x_{00} 到 x_{60} 的最短路。

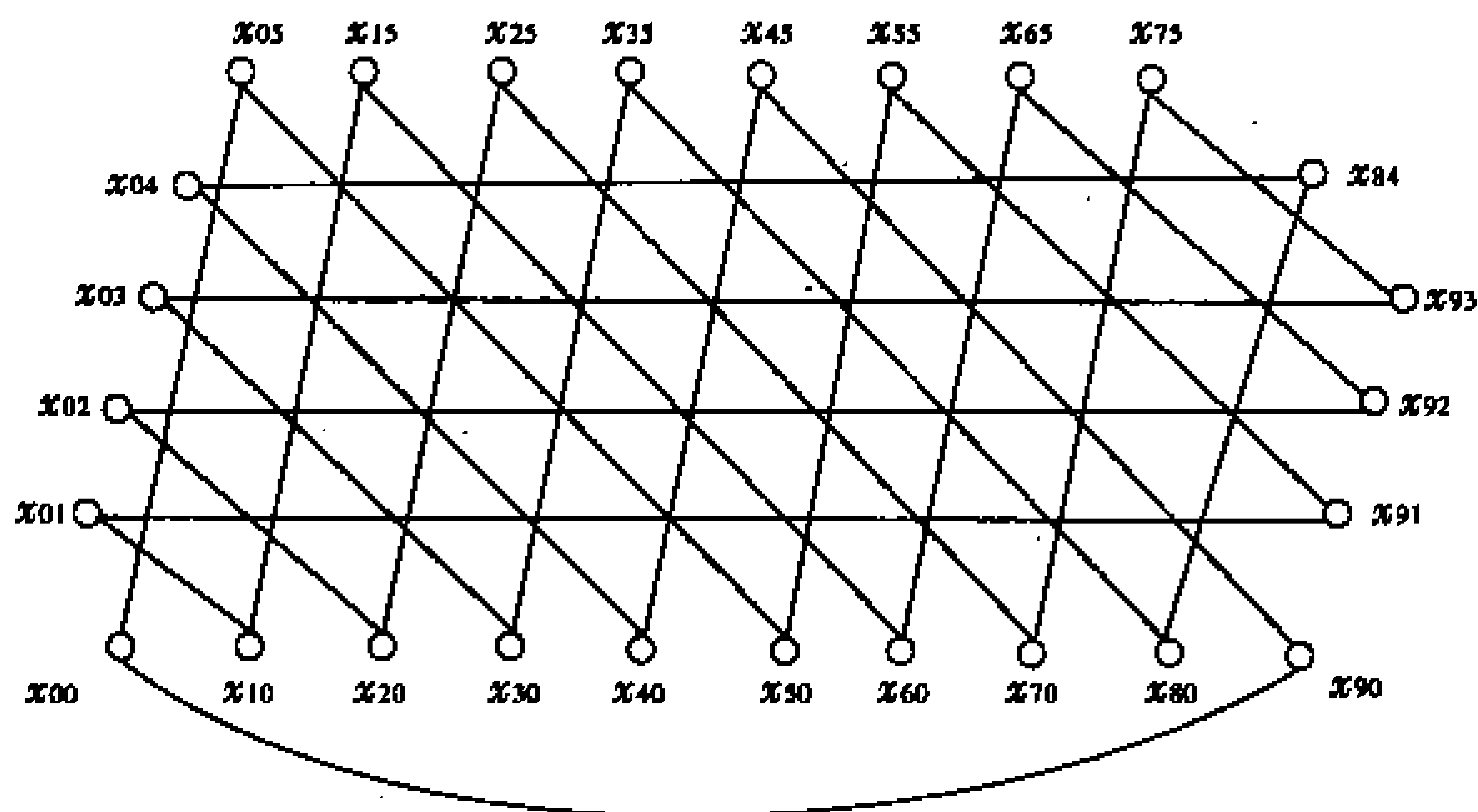


图6 G

7 **解** 我们先来建立一个图 G。图 G 中的顶点 x_i 代表这个图形的区域 $x_i(i=1,2,3,4,5,6)$。顶点 x_i 与 x_j 之间连接的边数等于区域 x_i 与 x_j 公共线段的数目，如图 7 所示。这样建立的图 G 中的每一条边对应这个图形的一条线段。存在满足条件的折线当且仅当 G 中存在一条 Euler 路或 Euler 回路。

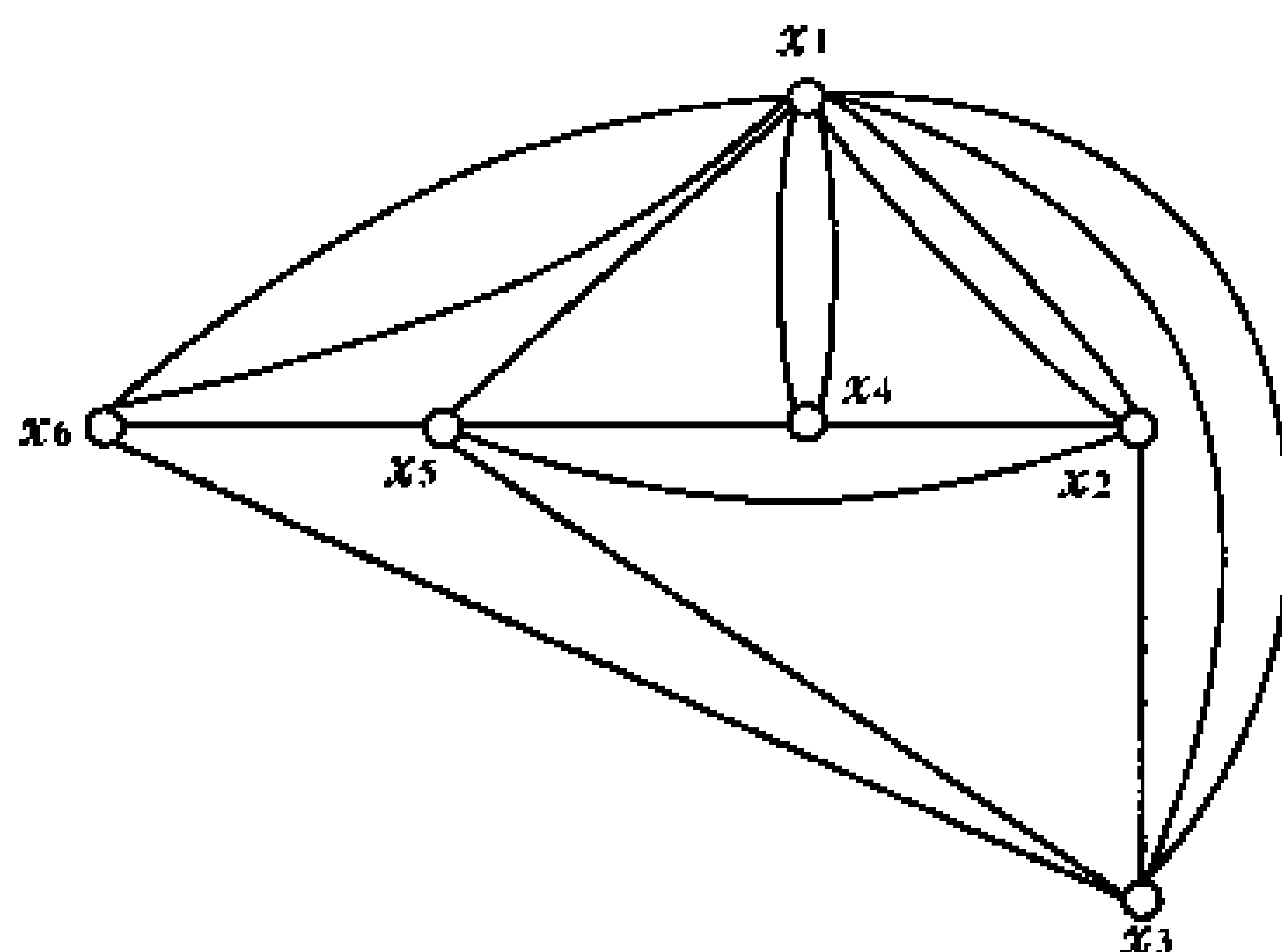


图7 G

8 **解** 首先建立图 $G=\langle V,E\rangle$，G 的 n 个顶点代表 n 个人。

两个不同的顶点 v_i 和 v_j 之间连接的边数等于这两个点所对应的两个人同时给出相同问题解的个数。G 的每一条边对应一个问题的解，每个顶点的度为 4。因而研究所一共收到 $2n$ 个不同问题的解。

不妨设 G 是连通图，由已知得 G 是 Euler 图，设 $C=v_0e_1v_1e_2v_2e_3\cdots e_{2n}v_0$ 是 G 的一条 Euler 回路，则不同于 v_0 的每个顶点在 C 内部各出现两次，而 v_0 有 C 内部出现一次。现在我们沿着 C 交错地给边染上红、蓝两色，则除 v_0 外，其他每个顶点有两条红边和两条蓝边与之关联。而对于 v_0，由于 G 有偶数条边，故也有两条红边和两条蓝边与 v_0 关联。则我们只要将红边所对应的 n 个问题的解安排在第一次发表，蓝边所对应的 n 个问题的解安排在第二次发表。

10 **解** 用结点代表人，两人是朋友时相应结点间连一边，得到一个无向图 $G=\langle V,E\rangle$。对 $\forall u,v\in V$，有 $d(u)\geqslant 10, d(v)\geqslant 10$，因而 $d(u)+d(v)\geqslant 20$，所以 G 中存在哈密顿回路。

11 **解** 用 7 个结点表示这 7 个人，若两人能交谈（会讲同一种语言），就在这两个结点间连一条边，得到图 G，如图 8 所示。图 G 中存在哈密顿回路。

12 **解** 用 6 个结点代表 6 个人，若两个人能合作，就在这两个人之间连一条边，得到无向图 $G=\langle V,E\rangle$，对于 $\forall u\in V$，有 $d(u)\geqslant 3$，因此 G 是哈密顿图。

13 **解** 构造一个具有 9 个结点的有向图，结点集为 $\{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$。从结点 $\alpha_1\alpha_2$ 到结点 $\alpha_2\alpha_3$ 有一条有向边记为 $\alpha_1\alpha_2\alpha_3$，其中 $\alpha_i\in\{a,b,c\}, 1\leqslant i\leqslant 3$。每一结点有三条有向边以它为起点，另有三条有向边以它为终点，如图 9 所示。每个结点的出度和入度都等于 3，并且对任意两个结点 $\alpha_1\alpha_2$ 和 $\beta_1\beta_2$，必有一条从结点 $\alpha_1\alpha_2$ 出发到结点 $\alpha_2\beta_1$ 再到结点 $\beta_1\beta_2$ 的

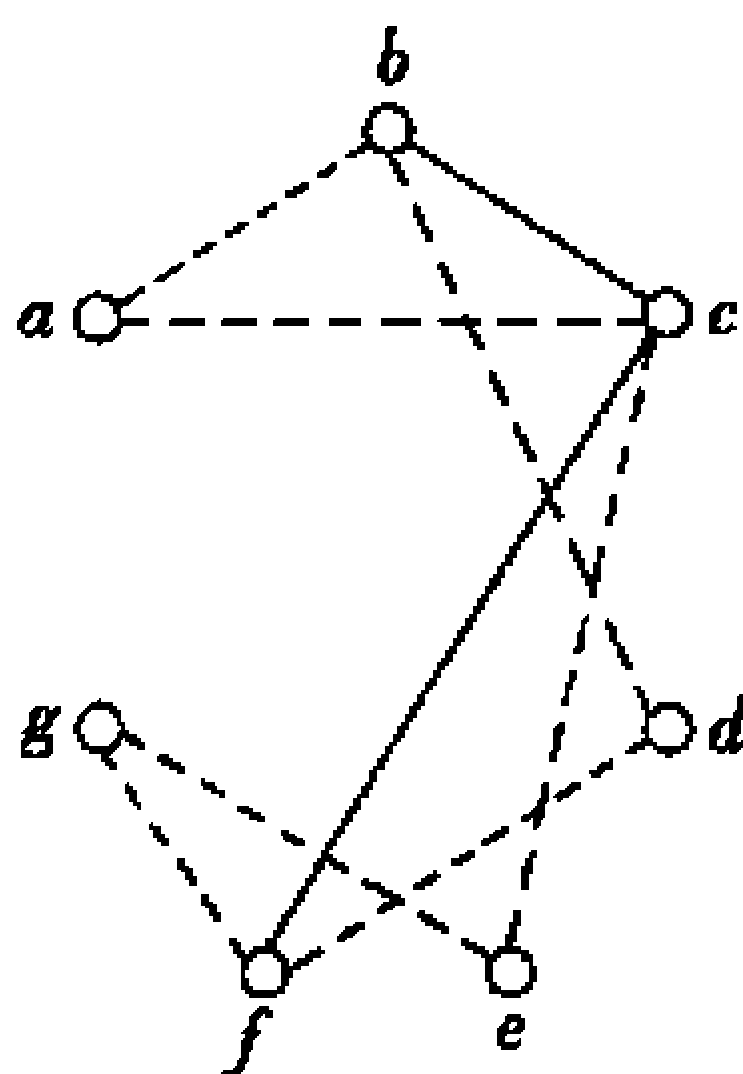


图 8 G

路，所以这个有向图是强连通的。此图有一条欧拉回路 $e_0e_1e_3e_{11}e_7e_{21}e_{10}e_4e_{14}e_{16}e_{22}e_{13}e_{12}e_9e_2e_8e_{26}e_{25}e_{23}e_{15}e_{20}e_6e_{19}e_5e_{17}e_{24}e_{18}$，此欧拉回路所对应的由 9 个 a，9 个 b，9 个 c 所对应的圆形排列 *aaabacbabbcbbbaacccbcacabcc*，即为所求。

14 **证** 因为 $|X|\neq|Y|$，所以不妨设 $|X|<|Y|$，则有

$$\omega(G-X)=|Y|>|X|$$

15 **证** 假设 G 不是汉密尔顿图，则存在结点 $u_1,u_2\in V$，使得 $d(u_1)+d(u_2)\leqslant n-1$。在 $G-\{u_1,u_2\}$ 中，结点数为 $|V|-2=n-2$，故它的边数 $\leqslant\frac{1}{2}(n-2)(n-3)$。$G$ 中的边数 $m\leqslant\frac{1}{2}(n-2)(n-3)+(n-1)<\frac{1}{2}(n-2)(n-3)+n=C_{n-1}^2+2$，与假设矛盾，因此 G 是汉密尔顿图。

16 **解** 每个子立方体作为一个结点，对结点用两种不同的标记，相邻的立方体用不同的标记，如图 10 所示。然后，根据相邻子立方体之间可以通导线的理由，对相应的结点连边即得图 11。显然 G 是一个二分图，其结点集为 X 和 Y，且 $|X|=14$，$|Y|=13$，立方体 8 个角对应的结点为 1，3，7，9，19，21，25，27，它们都属

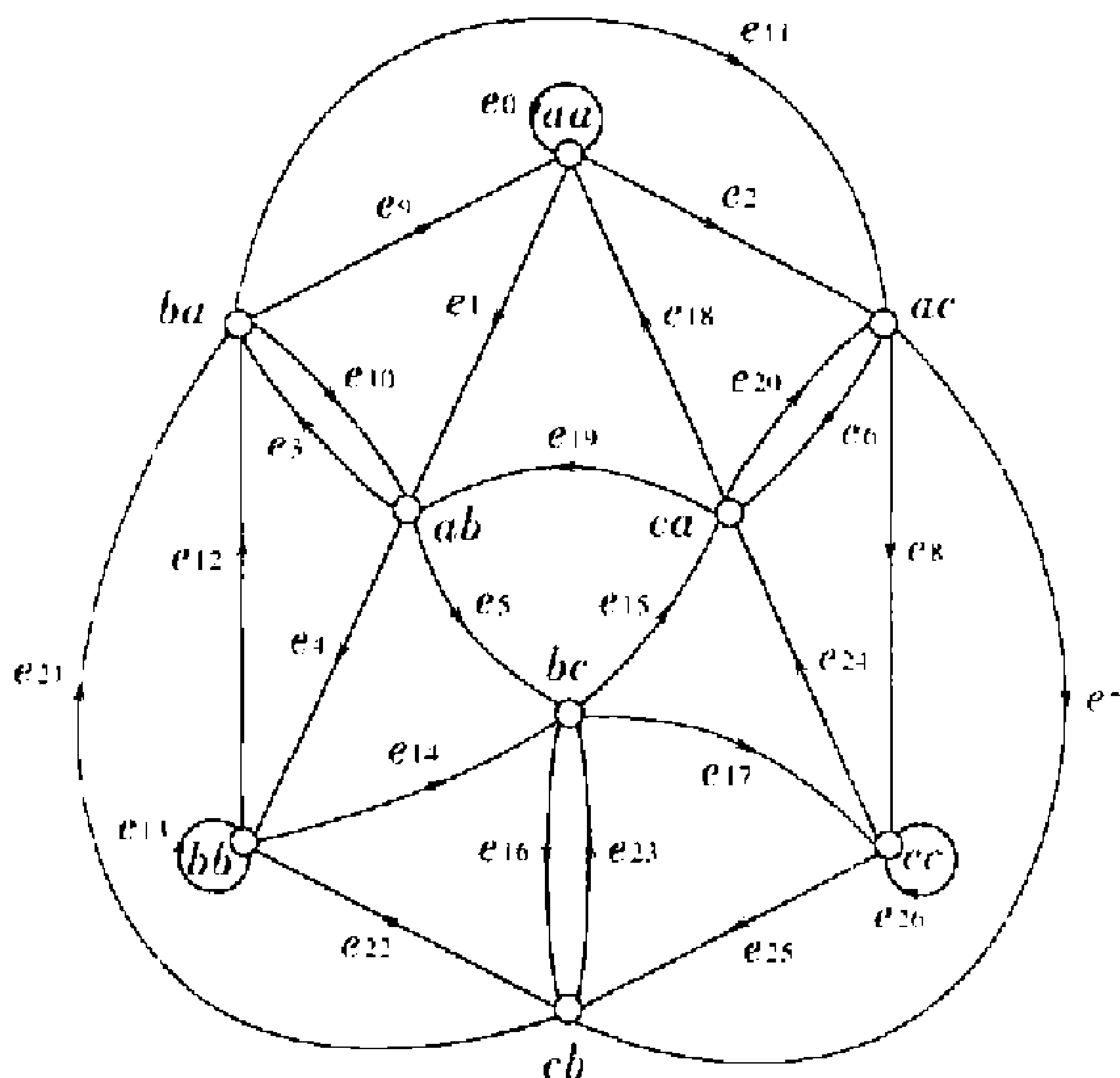


图 9

于 X,中心子立方体对应的结点为 $14 \in Y$。因此,若从某一角出发穿导线最后可终止于中心,则从起点到终点连边后所得的二分图 G' 中仍是 $|X| > |Y|$,而 G' 为哈密尔顿图,这与题中的结果相矛盾。因此,该题中规定的通导线方法是不可能实现的。

17 **解** 用 n 个点表示 n 个人,每两个人都有可能相邻就座,因此每两个点之间都可连线。这样就可得到一个具有 n 个结点的完全图 K_n。那么,K_n 中一个哈密尔顿回路就是一次晚餐的就座方法。可见,晚餐最多能进行的次数就是 K_n 中无公共边的哈密尔顿回路的个数。

K_n 中共有 $\frac{n(n-1)}{2}$ 条边,每个哈密尔顿回路有 n 条边,因此,边不相重的哈密尔顿回路最多有 $[\frac{n-1}{2}]$ 条。

对于 n 为奇数的情况,做出图 12(a)的图形。显然,(1,2,3,

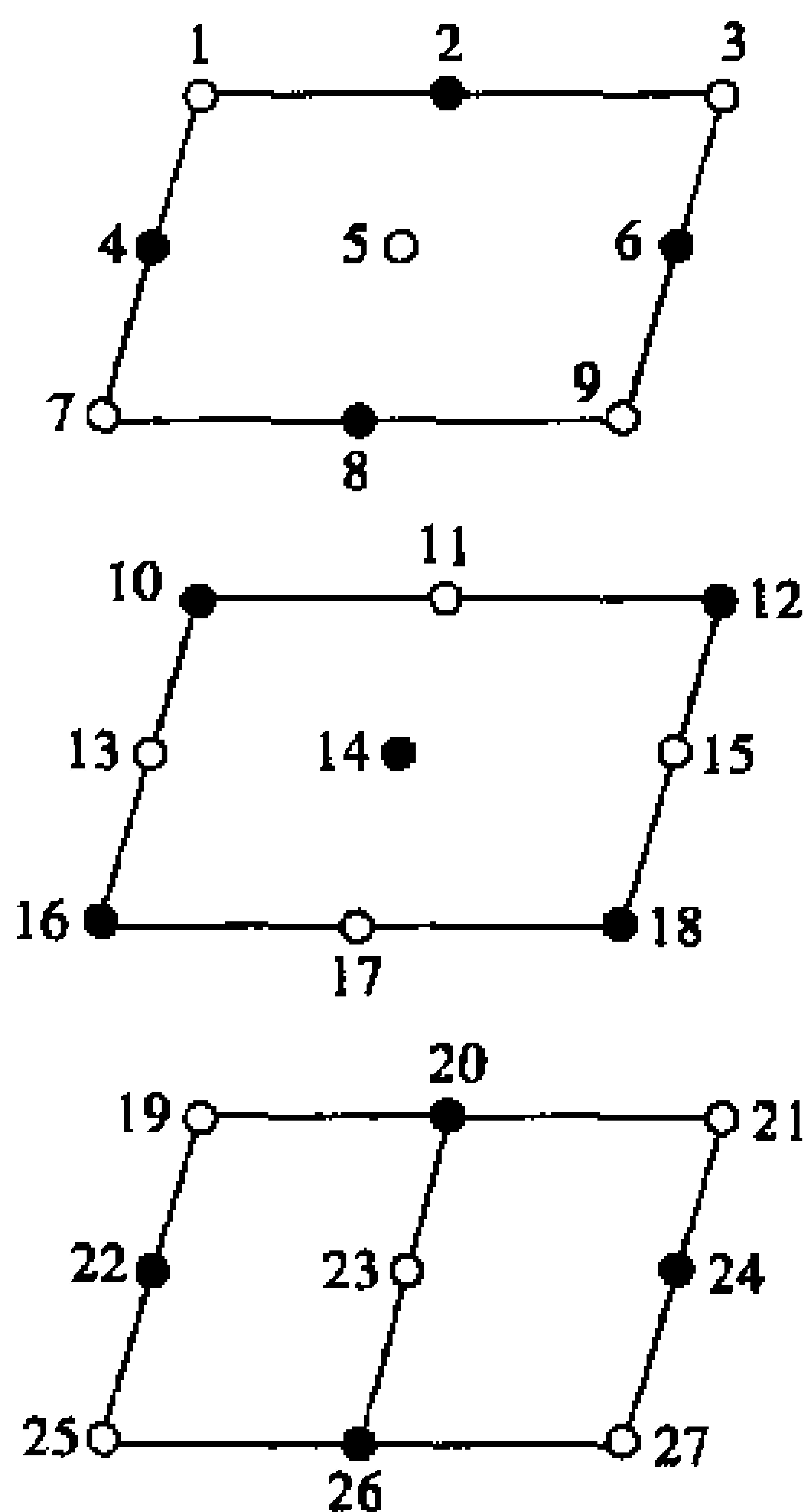


图 10

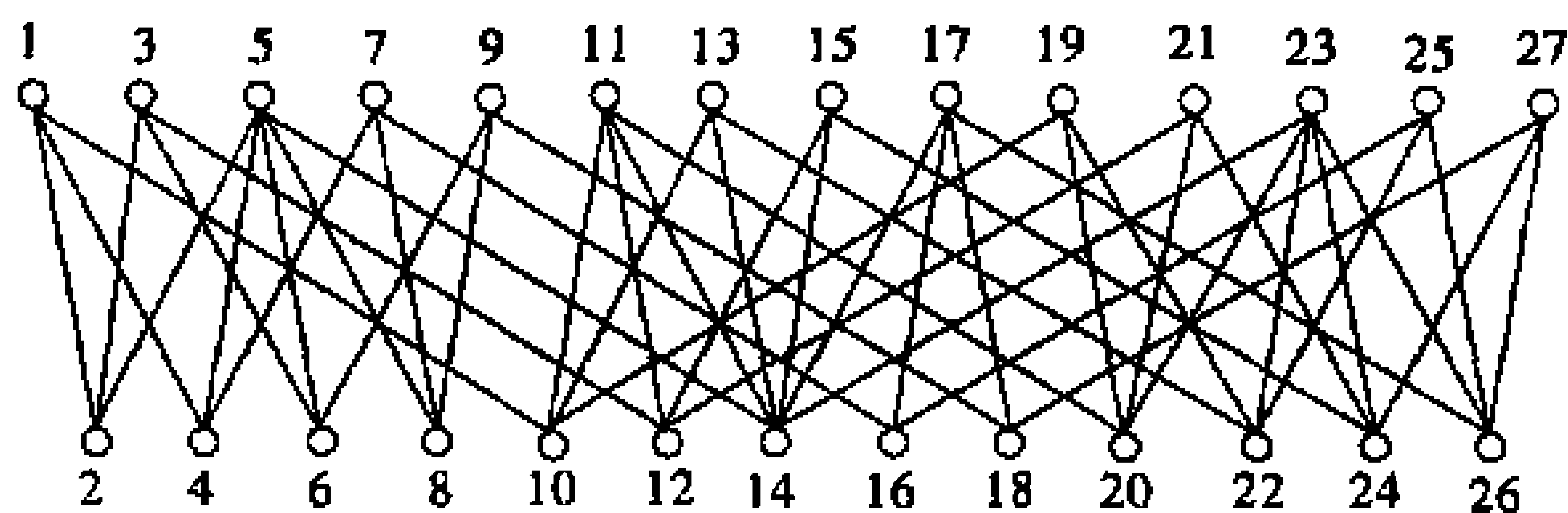


图 11

4, …, $n-3, n-2, n-1, n$, 1)是 K_n 中的一条哈密尔回路。

现将圆周上的点的编号依次顺时针旋转$\frac{360°}{n-1}$, $2\times\frac{360°}{n-1}$, …,

$\frac{n-3}{2}\times\frac{360°}{n-1}$，可得图 12(b)，(c)，(d)所示。

对于 n 为偶数的情形，可做出图 12(e)图形，即对 $n-1$ 是奇数的图形，并在中间添加一个结点 n。因此，可类似地得$\left[\frac{n-1}{2}\right]$个边不相重的哈密尔顿回路。

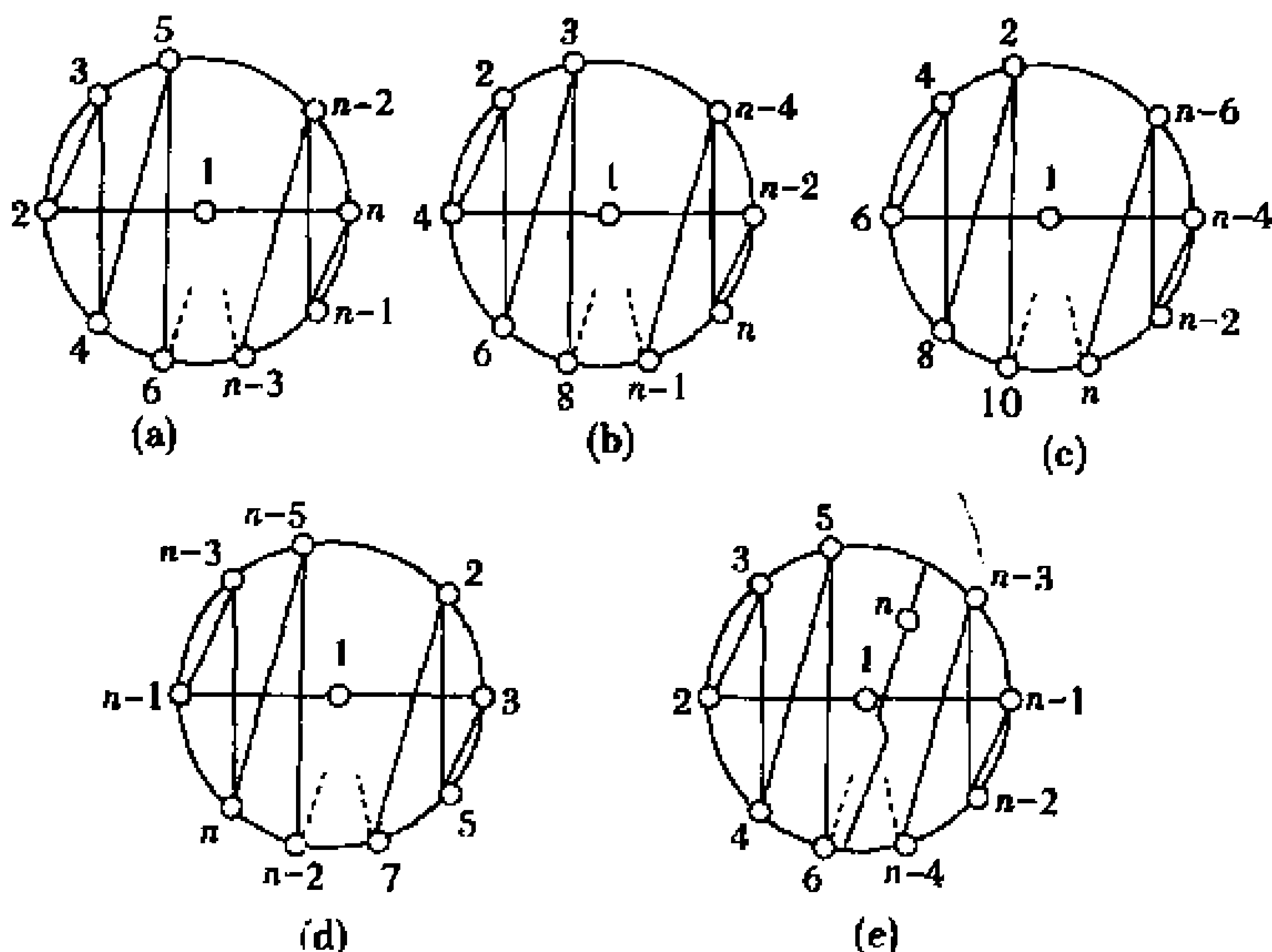


图 12

习 题 三

1. **证** 设 G 的 k 棵树分别为 $T_1,T_2,\cdots,T_k$，且 T_i 有 n_i 个顶点，m_i 条边，则 $m_i=n_i-1$。故 $m=\sum_{i=1}^{k}m_i=\sum_{i=1}^{k}(n_i-1)$

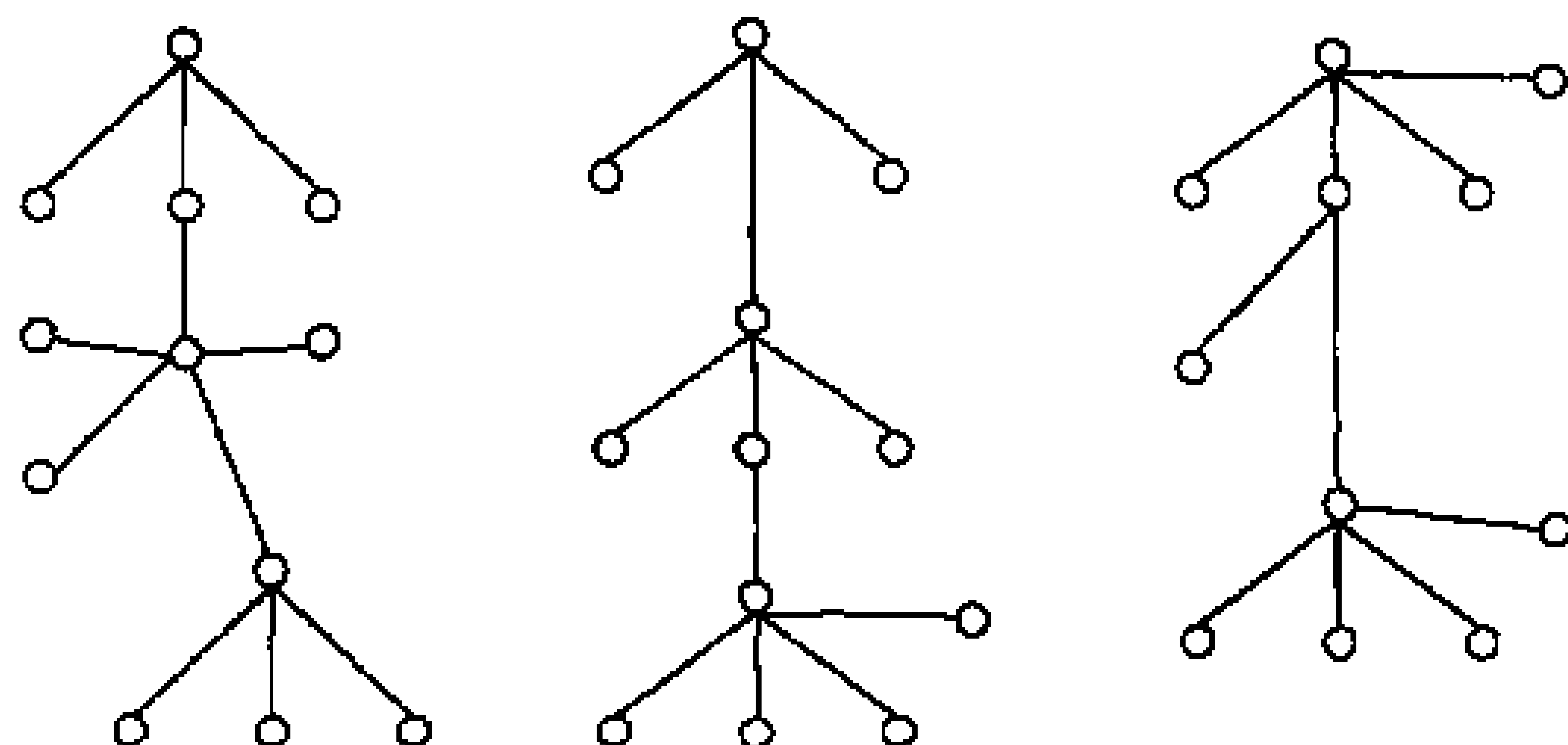
$= \sum_{i=1}^{k} n_i - k = n - k$。

2 证 设 G 有 $k(k \geqslant 1)$ 个连通分支 $T_1, T_2, \cdots, T_k$，假设 G 中无圈，则 $T_i(1 \leqslant i \leqslant k)$ 中也无圈，因而 T_i 均为树。设 T_i 有 n_i 个顶点，m_i 条边，则 $m_i = n_i - 1$。于是，$m = \sum_{i=1}^{k} m_i = \sum_{i=1}^{k} (n_i - 1) = n - k(k \geqslant 1)$。

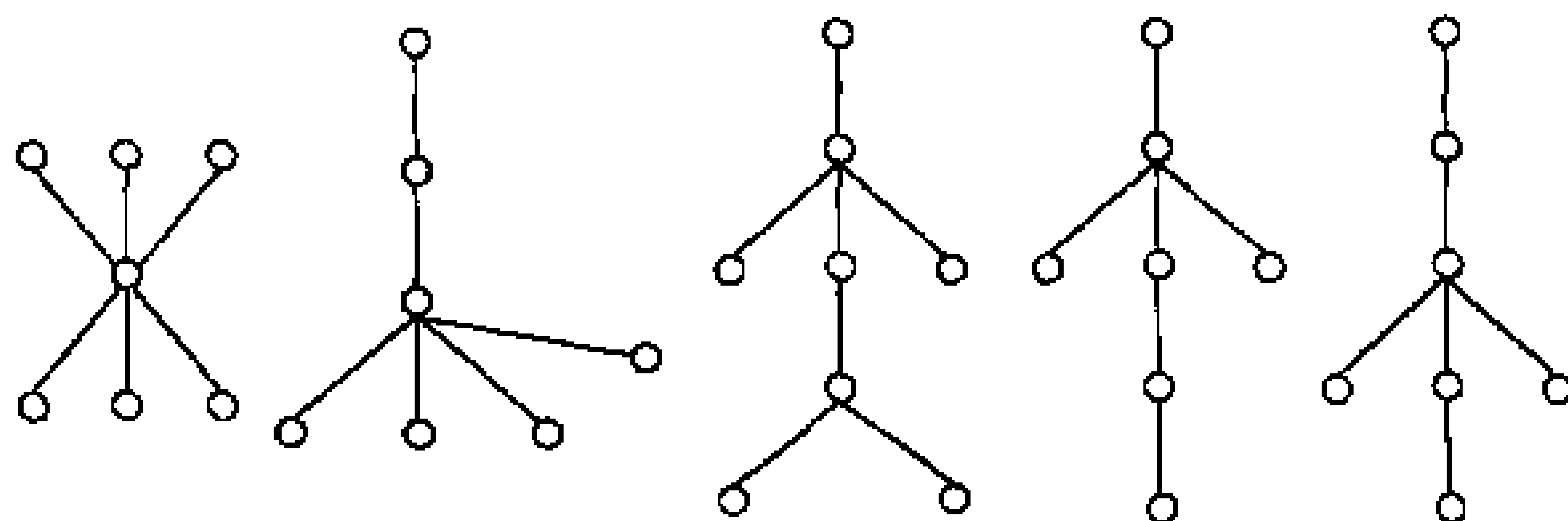
从而 $m < n$，这与 $m \geqslant n$ 矛盾。故 G 中有圈。

3 解 5

4 解

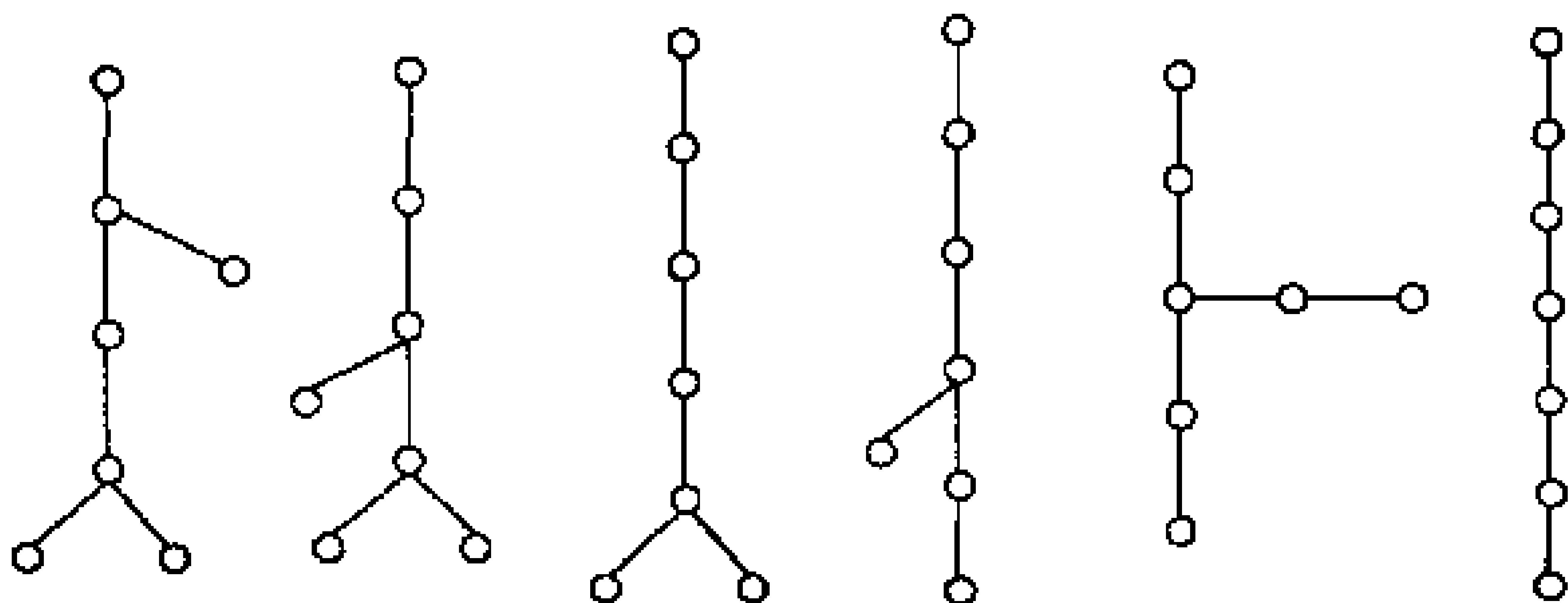


5 解



6 解 9

7 解 999



9　**解**　3,16,n^{n-2}

10　**证**　令 T 是一棵树。根在 T 的某个任意点,令 V_1 是偶数层结点的集合,V_2 是奇数层结点的集合,因为每个边与 V_1 中一个点和 V_2 中一个点相关联。

11　**解**　8;6,6,7,7,7,8,8,9

16　输入　图 $G=\langle V,E\rangle$,具有 n 个顶点

　　输出　true,如果 G 连通

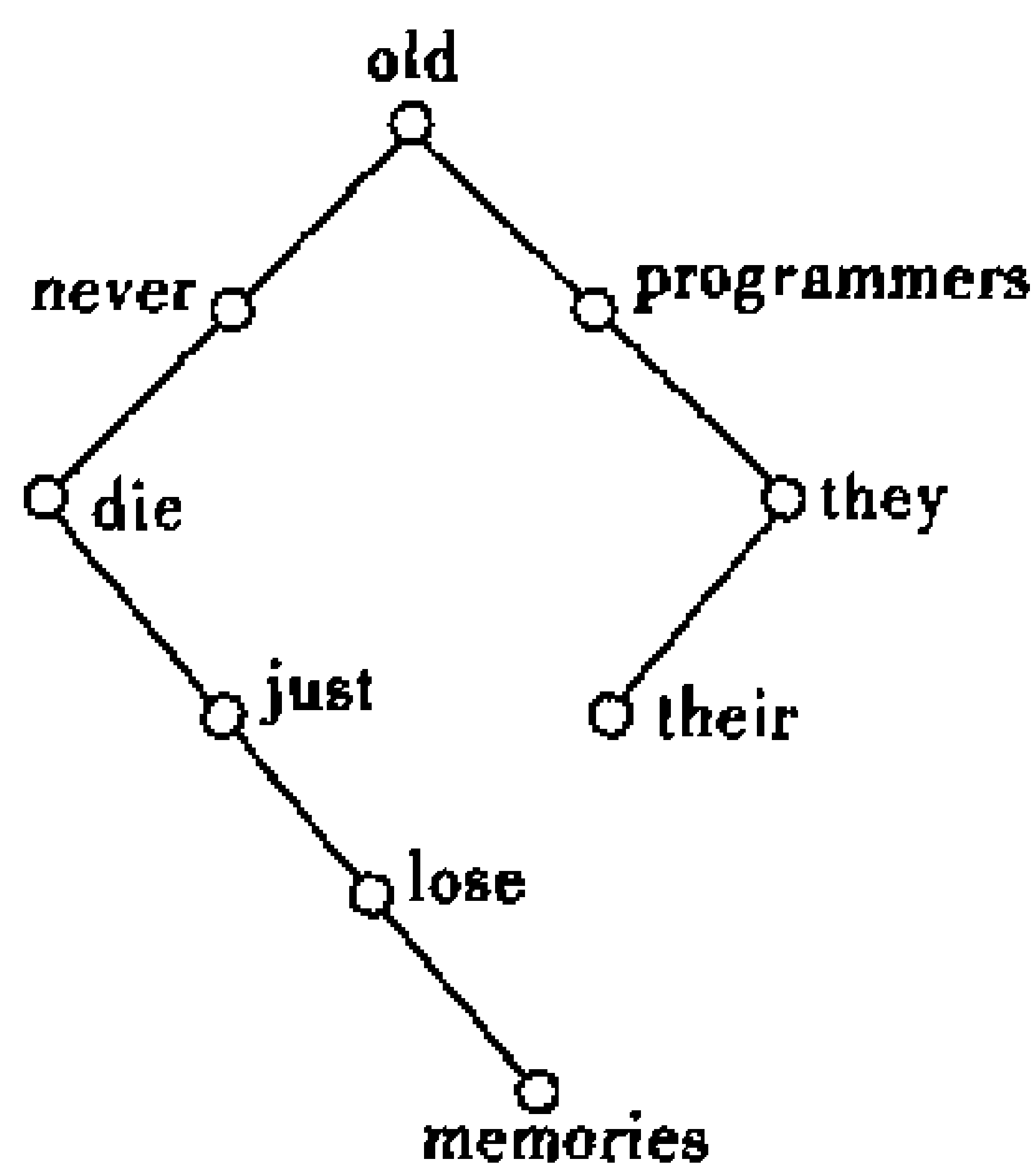
　　　　　false,如果 G 不连通

```
  procedure ected⟨V,E⟩
T:=bfs⟨V,E⟩
{T=⟨V′,E′⟩是由 bfs 返回的生成树}
  if  |V′|=n  then
return(true)
else
return(false)
end  is-connected
```

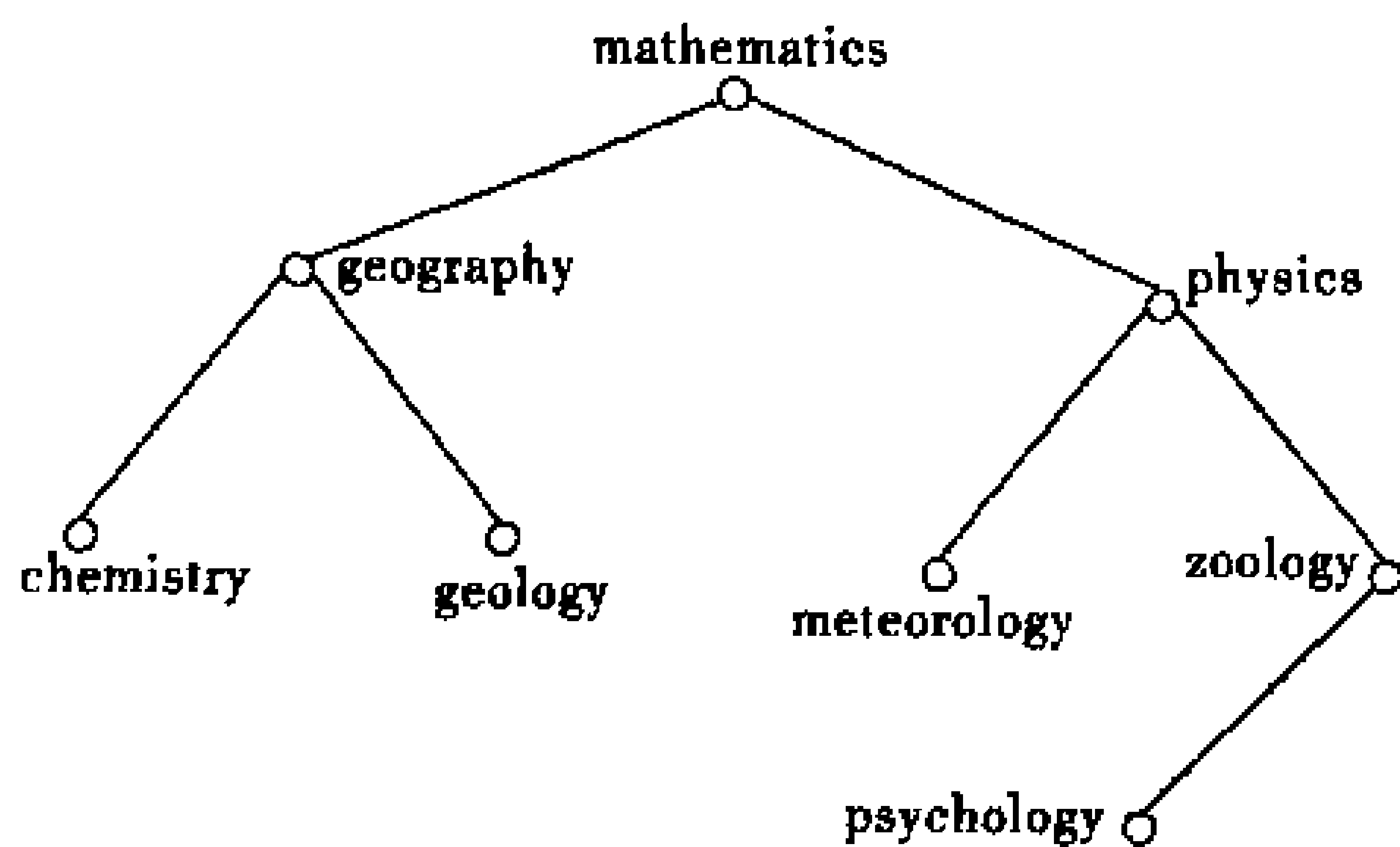
18　**证**　假设 G 有两个最小生成树 T_1 和 T_2,则存在一条边 x 在 T_1 中而不在 T_2 中,存在一条边 y 在 T_2 中而不在 T_1 中,使得 $T_3=T_1-x+y$ 和 $T_4=T_2-y+x$ 都是生成树。因为 x 与 y 具有不同的权。

20　**解**　$t-1$

21

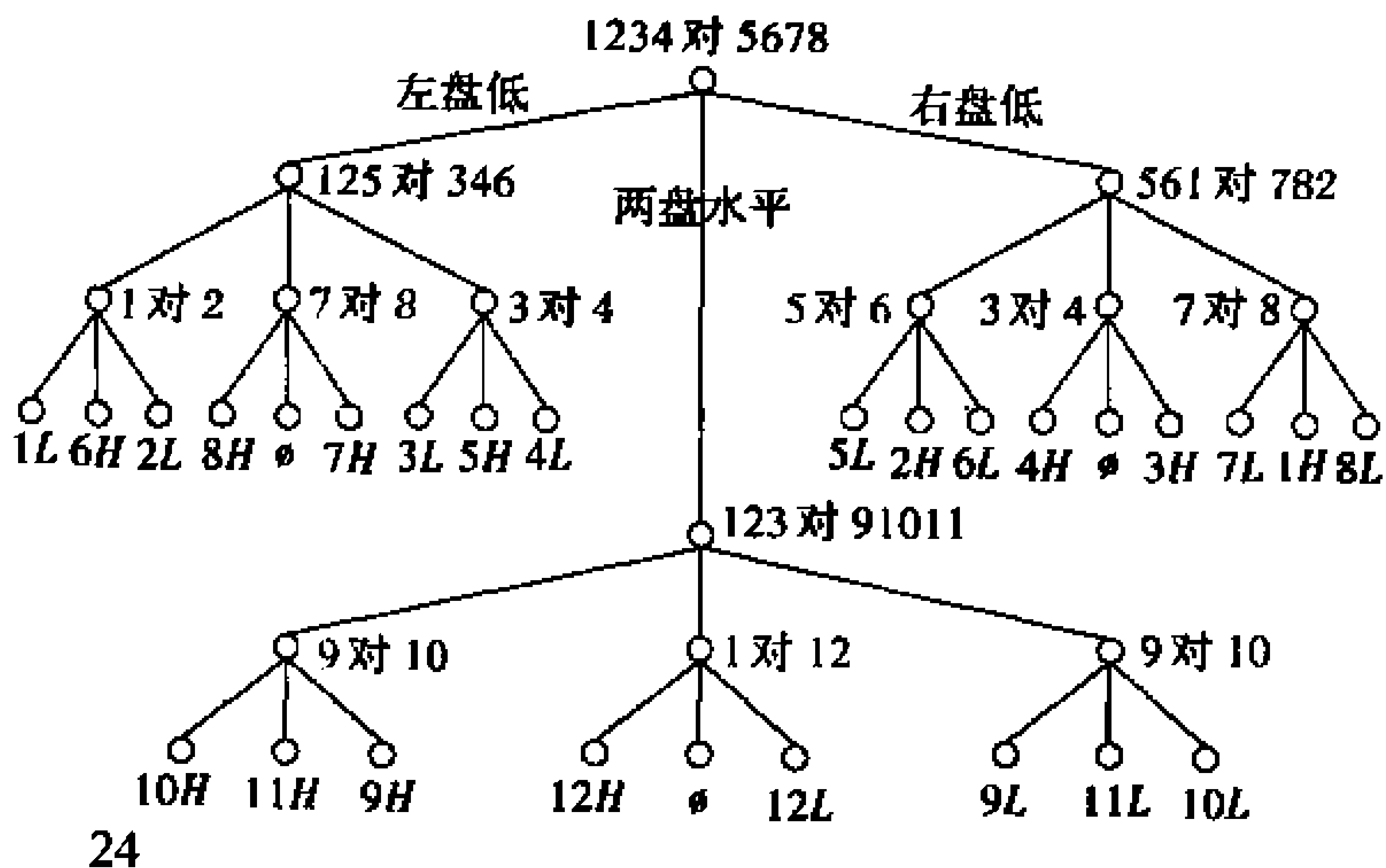

old
never
programmers
die
they
just
their
lose
memories

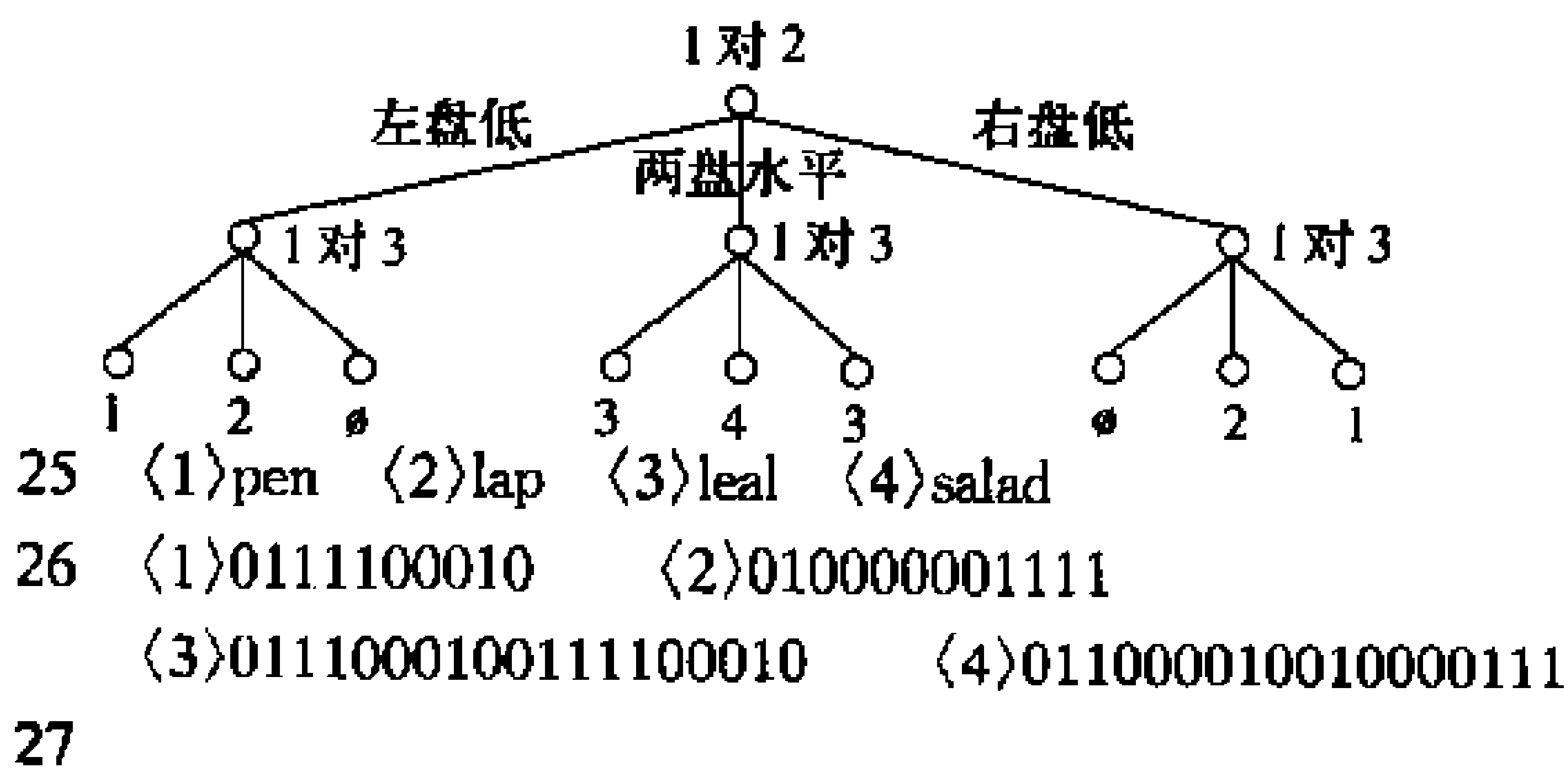
22


mathematics
geography
physics
chemistry
geology
meteorology
zoology
psychology

23

24



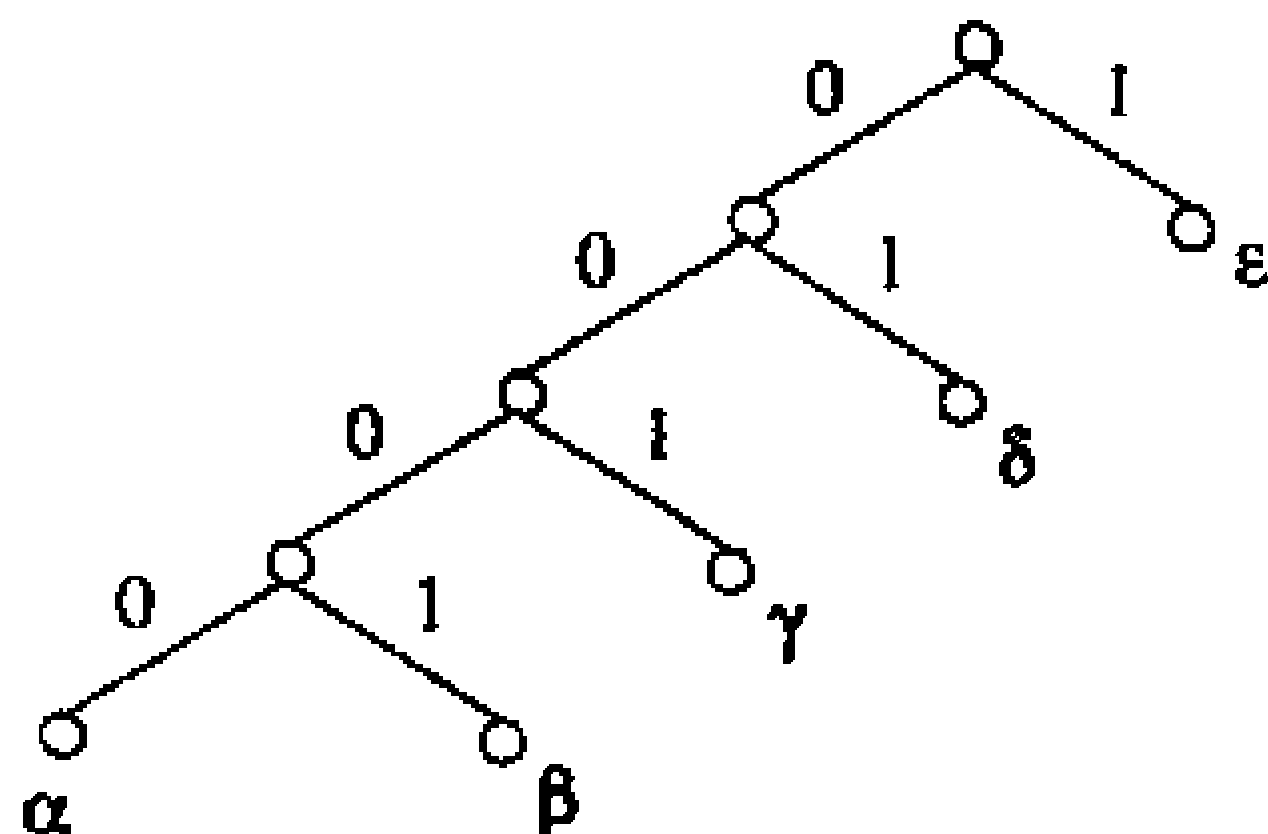


25 〈1〉pen 〈2〉lap 〈3〉leal 〈4〉salad

26 〈1〉0111100010 〈2〉010000001111

〈3〉011100010011110010 〈4〉011000010010000111

27



习 题 四

1 **解** 8

2 **解** 平面性的

5 **证明** 只要找出它们各自的一个平面表示(平面嵌入)即可。图 21 分别给出了它们的平面表示。

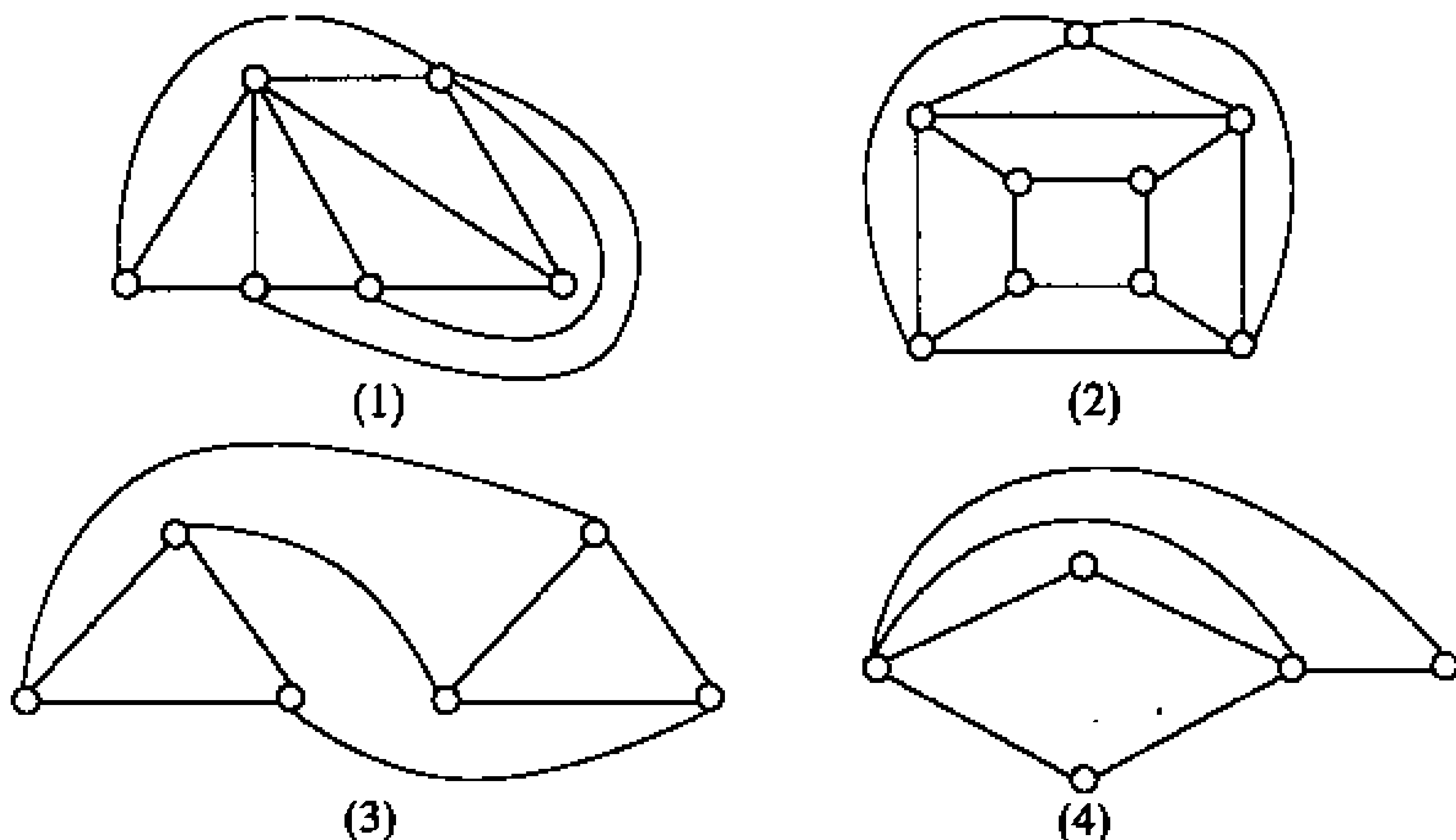


图 21

6 **证明** 若 G 为非连通图,设 G 有 $k(k\geqslant 2)$ 个连通分支 $G_1,G_2,\cdots,G_k$,取 v_{11} 在 G_1 中,v_{21} 在 G_2 中,在 G 上加边 $\{v_{11},v_{21}\}$,所得新图 $G+\{v_{11},v_{21}\}$ 还是平面图,这与 G 为极大平面图矛盾。

7 **证明** 用反证法。若 $\delta(G)\leqslant 2$,分情况讨论:

(1)$\delta(G)=1$,G 中存在顶点 v_1,$d(v_1)=1$。设与 v_1 相邻的顶点为 v,并设 v 除与 v_1 相邻外,还依次地与 $v_2,\cdots,v_s$ 相邻,因 v_1 不与 v_2 相邻,加边 $\{v_1,v_2\}$ 不破坏平面性,这与 G 为极大平面

图矛盾，所以 $\delta(G)\geqslant 2$。

(2) $\delta(G)=2$，G 中存在顶点 v_1，$d(v_1)=2$，设与 v_1 相邻的顶点为 v 和 v_2，这时 v 与 v_2 相邻，否则加边 $\{v_1,v_2\}$ 不破坏平面性，这与 G 为极大平面图矛盾。设 v 除与 v_1,v_2 相邻外，还依次地与 $v_3,v_4,\cdots,v_t$ 相邻，在 G 中 v_1 不能与 v_t 相邻(因为 $d(v_1)=2$)，加边 $\{v_1,v_t\}$ 不破坏平面性，这又矛盾于 G 为极大平面图，所以 $\delta(G)\geqslant 3$。

8 **证明** 因为 G 为极大平面图，所以 G 是连通的，由欧拉公式可知 $r=\varepsilon-v+2$ (1)

设 G 的平图各面边界长度之和为 l，G 的平图的每个面的边界之长均为 3，所以

$$3r=l=2\varepsilon \qquad (2)$$

将(1)代入(2)，经整理得 $\varepsilon=3v-6$。

9 **证明** 用反证法。设 G 为非连通的，具有 $k\geqslant 2$ 个连通分支 $G_1,G_2,\cdots,G_k$，设 G_i 的顶点数为 n_i，边数为 m_i，$i=1,2,\cdots,k$。

若存在 $n_j=1$，则 k 必为 2，因为只有此时 G 为一个平凡图并上一个 K_6 才能使其边数为 15，可是 K_6 不是平面图(它的子图 K_5 已不是平面图)，这矛盾于 G 为平面图这个事实，所以不存在 $n_j=1$。

若存在 $n_j=2$，G_j 中至多有一条边(因为 G 为简单图)，另外 5 个顶点构成 K_5 时边数最多，但充其量为 10 条边，这与 G 有 15 条边矛盾。

综上所述，n_i 必大于等于 3，$i=1,2,\cdots,k$。

$$m_i\leqslant 3(n_i-2)=3n_i-6,\ i=1,2,\cdots,k。$$

求和得 $\varepsilon\leqslant 3v-6k$ (1)

将 $v=7$，$\varepsilon=15$ 代入(1)得 $15\leqslant 21-6k$

从而得 $k\leqslant 1$，这与 $k\geqslant 2$ 矛盾。

10 **证明** (1) G 中任何可平面子图的边数，不可能超过 v 个顶点的极大平面图的边数。而 $v(v\geqslant 3)$ 个顶点的极大平面图的边数为 $3v-6$，因而，当 G 有 ε 条边时 $\theta(G)\geqslant\left\{\frac{\varepsilon}{3v-6}\right\}$。

(2)对于任意的正整数 x,y，有

$$\left\{\frac{x}{y}\right\}=\left[\frac{x+y-1}{y}\right],$$

于是 $$\theta(G)\geqslant\left\{\frac{\varepsilon}{3v-6}\right\}=\left[\frac{\varepsilon+3v-7}{3v-6}\right]。$$

11 **解** 〈1〉在 K_5 中，顶点数 $v=5$，边数 $\varepsilon=10$，于是

$$\theta(K_5)\geqslant\left\{\frac{10}{3\times 5-6}\right\}=\left\{\frac{10}{9}\right\},$$

但是 K_5 存在着两个可平面的子图，它们的并为 K_5（如图 22(a)，(b)所示）。所以 $\theta(K_5)=2$。

〈2〉在彼得森图 G 中，$v=10,\varepsilon=15$，所以

$$\theta(G)\geqslant\left\{\frac{15}{24}\right\}=1。$$

可是因为 G 为非平面图，因而 $\theta(G)\geqslant 2$，又因为 G 为图 22(c)和(d)的并，因而 $\theta(G)=2$。

〈3〉图 22(e)和(f)两图的并为 4 维立方体图 Q_4，因而 4 维立方体图的厚度也为 2。

〈4〉2

12 **解** (1) 1 (2) 3 (3) 9 (4) 1 (5) 2 (6) 4 (7) 6

17 **解** 若 n 是偶数则为 3，若 n 是奇数则为 4。

20 **解** 5

21 **证明** 设 G^* 是 G 的对偶图，图为 $G\cong G^*$，所以 G 必为连通的平面图。而 $v^*=r,\varepsilon^*=\varepsilon,r^*=v$，于是 $v=v^*=r$，由欧拉公式知

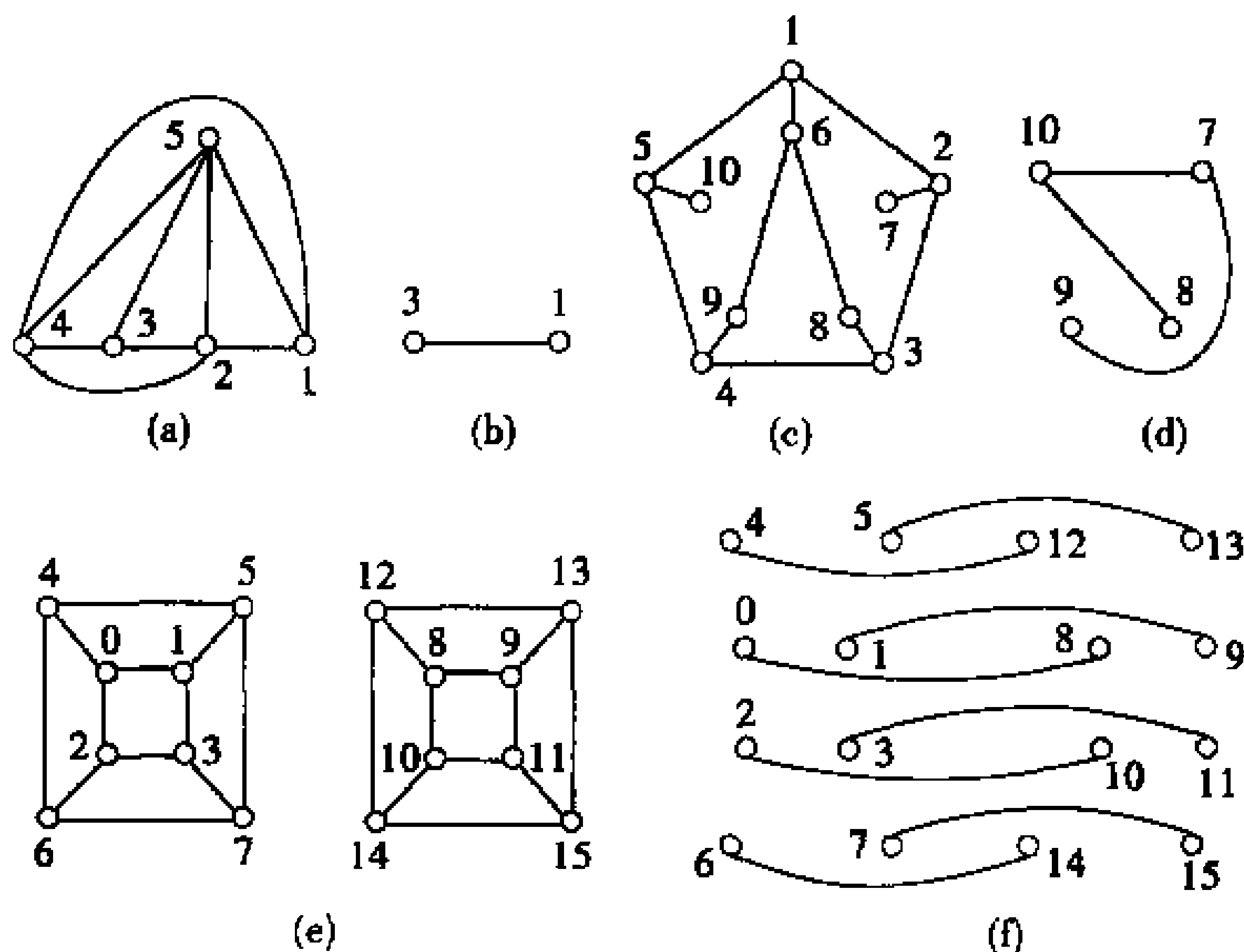


图 22

$$v-\varepsilon+r=2=v-\varepsilon+\varepsilon$$

所以　$\varepsilon=2v-2$。

22　**证明**　W_n 中每个三角形内含 W_n^* 中的一个顶点，W_n 的无限面中含 W_n^* 的一个顶点 v_0^*，各三角形内的顶点依次相邻形成 W_n^* 中的一条长为 $n-1$ 的回路，回路上每个顶点均与 v_0^* 相邻，形成 W_n^* 的 $n-1$ 个三角形围成的面，由轮图定义可知 W_n^* 是轮图。并且顶点数 v^* 与 W_n 的顶点数 v 相同，所以 $W_n^*\cong W_n$。

23　**证明**　显然 G^* 是连通图。设 v^* 为 G^* 中任意一顶点，v^* 处于 G 的面 r 中，由于 r 由偶数条边围成，所以 $d(v^*)$ 为偶数，由于 v^* 的任意性可知，G^* 为欧拉图。

24　**证明**　必要性，设 G^* 为 G 的对偶图，则 G^* 是连通的，只要证明 G^* 中每个顶点的度数均为偶数。因为 G 为二分图，因

此 G 中无奇数长度的圈，因而 G^* 中所有顶点的度数均为偶数，所以 G^* 是欧拉图。

充分性 因为 G^* 为欧拉图，所以 G^* 中每个顶点的度数都为偶数。所以 G 中无奇数长的圈，因而 G 必为二分图。

25 **解** 图 G 是否存在所要求的闭折线等价于 G 的对偶图 G^* 是否为欧拉图，亦或 G 的所有面的次数是否均为偶数。

习 题 五

1 **证** 设 M 是 G 的最大匹配，S 是非 M 饱和点集，则 S 的导出子图 $G[S]$ 是空图。由于 $\delta(G)>0$，所以 G 中存在 $|S|$ 条边集 E'，它的每条边都与 S 中点关联。显然，$M\cup E'$ 是 G 的边覆盖，因而 $\beta'\leqslant|M\cup E'|=\alpha'+(n-2\alpha')=n-\alpha'$。即

$$\alpha'+\beta'\leqslant n$$

另一方面，设 L 是 G 的最小边覆盖。令 $H=G[L]$，则 $V(H)=V(G)$。设 M 是 H 的最大匹配，S 为 H 中非 M 饱和点集。则 $H[S]$ 是空图，从而

$$|L|-|M|=|L-M|\geqslant|S|=n-2|M|,$$

即
$$|L|+|M|\geqslant n$$

又因为 H 是 G 的支撑子图，所以 M 也是 G 的匹配。

故
$$\alpha'+\beta'\geqslant|M|+|L|\geqslant n$$

2 **证** 设 G 是具有二划分 (V_1,V_2) 的二分图。我们对 G 的边数 $\epsilon\geqslant1$ 用归纳法来证明 $\beta(G)\leqslant\alpha'(G)$。若 $\delta(G)\leqslant1$，则结论显然成立。不妨设存在 $x\in V_1$，使 $\deg(x)\geqslant2$。取 $e_1,e_2\in E(G)$，使 $\varphi(e_1)=\{x,y\}$，$\varphi(e_2)=\{x,u\}$。设存在 $G-e_1$ 的点覆盖 S_1 使 $|S_1|=\beta(G)-1$ 且存在 $G-e_2$ 的点覆盖 S_2 使 $|S_2|=\beta(G)-1$。易见 $x\notin S_1$，$y\notin S_2$，$u\in S_1-S_2$。于是

$$|((S_1\cap S_2)\cap V_1)\cup((S_1\cup S_2)\cap V_2)|\geqslant\beta(G),$$

$$|(((S_1\cup S_2)\cup\{x\})\cap V_1)\cup((S_1\cap S_2)\cap V_2)|\geqslant\beta(G)$$

两式相加，得 $|S_1\cup S_2|+|S_1\cap S_2|+1\geqslant 2\beta(G)$

于是　$2\beta(G)-2=|S_1|+|S_2|\geqslant 2\beta(G)-1$.

这是不可能的. 于是 $\beta(G-e_1)=\beta(G)$ 或者 $\beta(G-e_2)=\beta(G)$. 不妨设 $\beta(G-e_1)=\beta(G)$. 于是由归纳假设.

$$\beta(G)=\beta(G-e_1)\leqslant\alpha'(G-e_1)\leqslant\alpha'(G)$$

3　**证**　设 G 是二分图，并且 $\delta(G)>0$，因而 $\alpha+\beta=\alpha'+\beta'$。又 $\alpha'=\beta$，于是 $\alpha=\beta'$。

6　**解**　(a)极小点覆盖 $\{a,c,e,g\}$，$\{b,c,d,e,g\}$，$\{b,d,e,f\}$ $\{b,c,d,f\}$

极大独立集 $\{b,d,f\}$，$\{a,f\}$，$\{a,c,g\}$，$\{a,e,g\}$

最小点覆盖 $\{a,c,e,g\}$，$\{b,d,e,f\}$，$\{b,c,d,f\}$

最大独立集 $\{b,d,f\}$，$\{a,c,g\}$，$\{a,e,g\}$

(b)极小点覆盖

$\{b,c,e,f\}$，$\{b,c,d,f\}$，$\{b,c,d,e\}$，$\{a,c,e,f\}$，$\{a,c,d,e\}$，$\{a,b,d,e\}$

极大独立集 $\{a,d\}$，$\{a,e\}$，$\{a,f\}$，$\{b,d\}$，$\{b,f\}$，$\{c,f\}$

最小覆盖

$\{b,c,e,f\}$，$\{b,c,d,f\}$，$\{b,c,d,e\}$，$\{a,c,e,f\}$，$\{a,c,d,e\}$，$\{a,b,d,e\}$

最大独立集 $\{a,d\}$，$\{a,e\}$，$\{a,f\}$，$\{b,d\}$，$\{b,f\}$，$\{c,f\}$

8　**解**　如图 23 所示。

10　**解**　(1)1　　(2)2　　(3)3

12　**证**　假设 G 是用 k 种颜色着色的并且独立数为 α. 因为每个颜色类必须是一个独立集，所以每个颜色类只有不超过 α 个元素。因此至多存在 $k\alpha$ 个顶点.

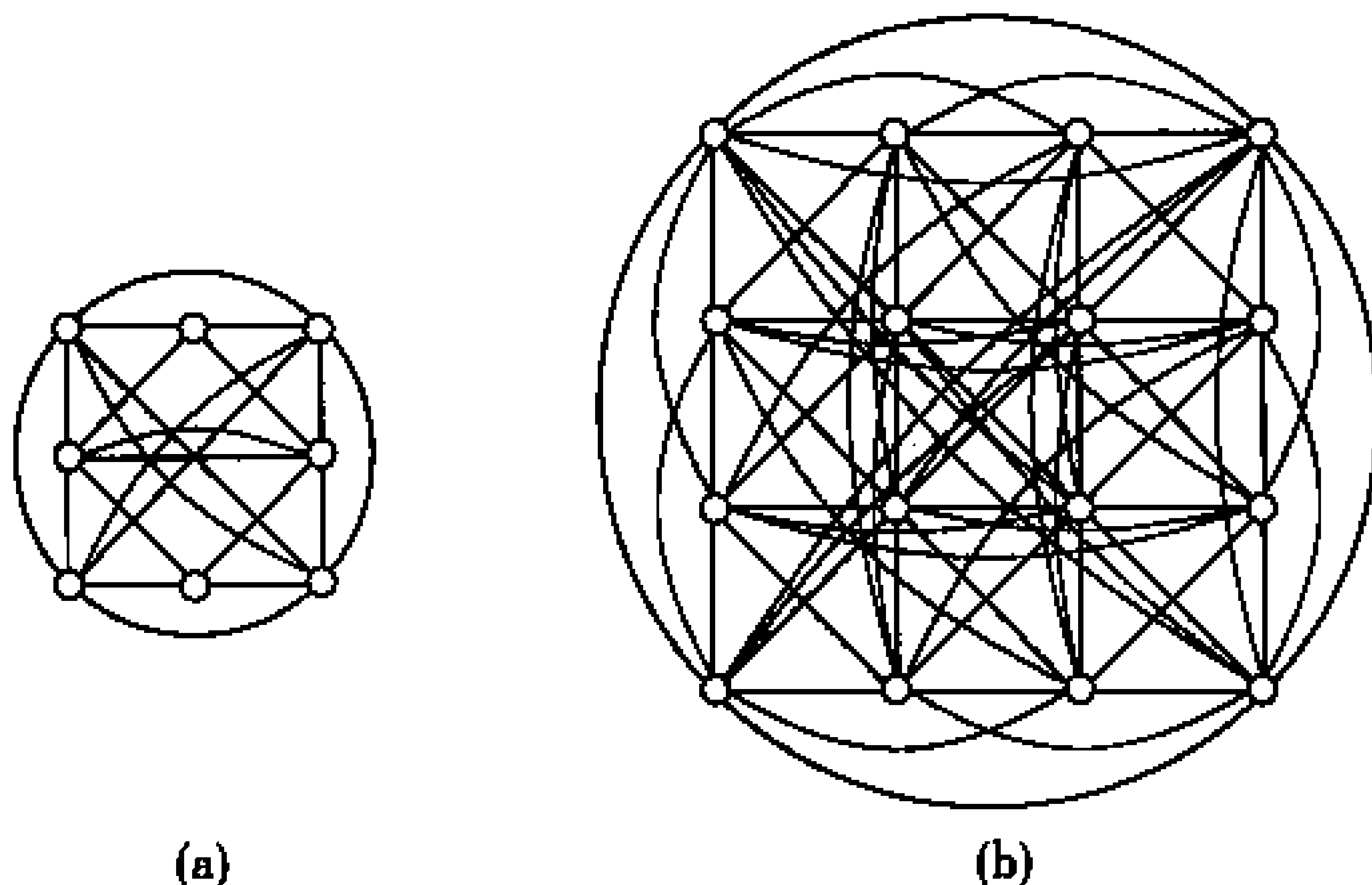


图 23

14 证 设 M 为树 T 中一完美匹配，下面证明 M 是 T 中唯一的完美匹配.

首先易知下面两个事实：

(1) T 中所有顶点都是 M 饱和点，因而树叶也必为 M 饱和点，且与树叶相关联的边均不相邻，且均在 T 的任何完美匹配中

(2)树 T 的顶点数 $n=2|M|$，所以 n 为偶数，于是 T 的边数 m 的奇数.

设 M_1 和 M_2 都是树 T 的完美匹配，并且它们是不同的，则 $M_1 \oplus M_2 \neq \emptyset$. 设 $M_1 \oplus M_2$ 的导出子图为 H. 由于 M_1 和 M_2 都是 T 的完美匹配，则 H 中顶点的度均为 2. 所以 H 中存在圈，这与 T 是树矛盾.

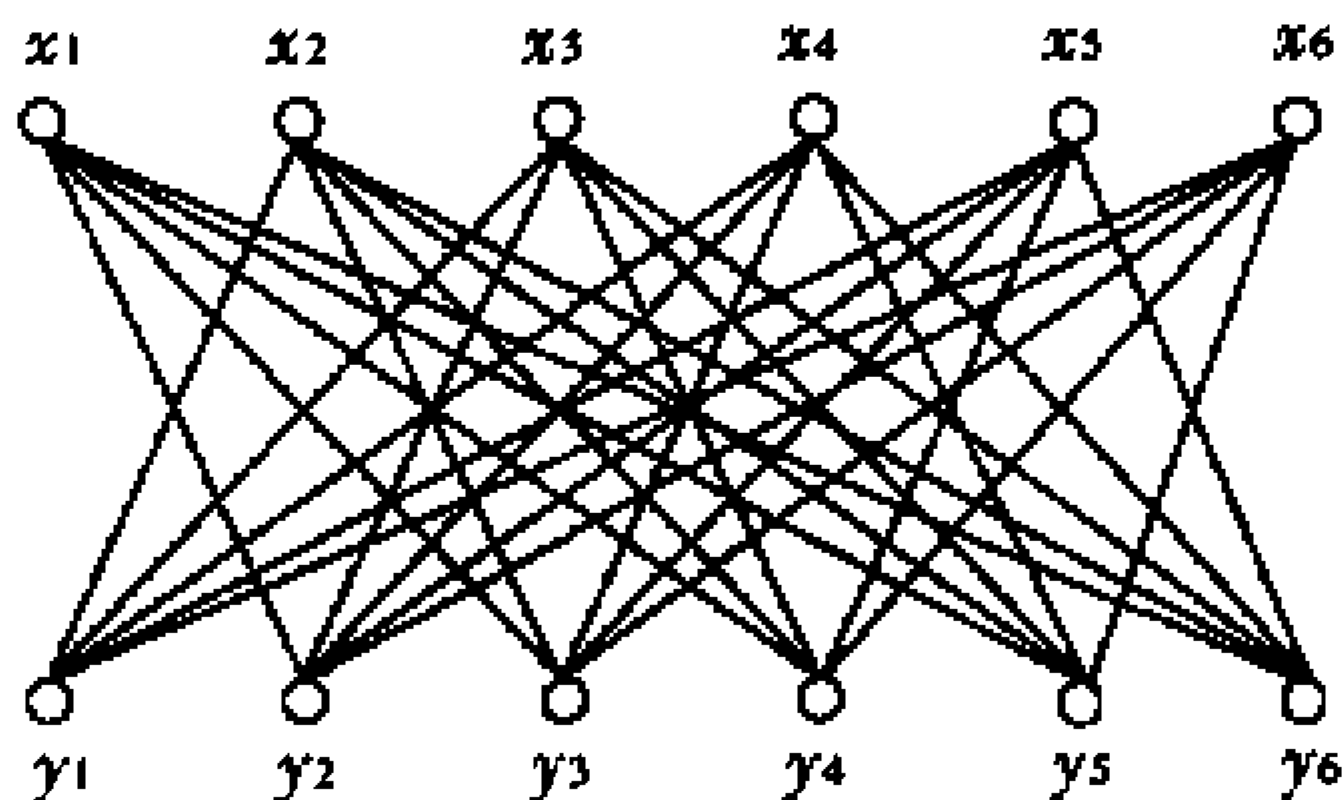
15 解 $2n-1, n$

17 解 取初始匹配 $M=\{x_1y_4, x_4y_1, x_5y_5\}$，求得完美匹配为 $\{x_1y_4, x_2y_1, x_3y_3, x_4y_2, x_5y_5\}$.

18 解 $J_2 \to J_3 \to J_4 \to J_5 \to J_1 \to J_6$

22 解 设信笺为 x_i，信封为 y_i，$i=1,2,3,4,5,6$，x_i 与 y_i

是相符的. 于是问题就转化为求图 G(如下图所示)



中有多少个不同的完美匹配,我们把这个数目记作 $\varphi(6)$, x_1 与 y_2 相配对时,完美匹配的个数等于从图 G 中删去顶点 x_1, y_2 后所得图 $G-\{x_1, y_2\}$ 中完美匹配的个数,这个数目记作 $\tau(5)$. 在 $G-\{x_1, y_2\}$ 中,若 x_2 与 y_1 相配对,则 $\tau(5)=\varphi(4)$;若 x_2 与 y_1 不配对,则 $\tau(5)=\varphi(5)$. 于是 G 中 x_1 与 y_2 相配对时,有 $\varphi(5)+\varphi(4)$ 个完美匹配. 由对称性, x_1 与 $y_j(3\leqslant j\leqslant 6)$ 相配对时,亦有 $\varphi(5)+\varphi(4)$ 个完美匹配,故 $\varphi(6)=5(\varphi(5)+\varphi(4))$,利用这个递推公式可求得 $\varphi(6)$.

23 **证明** 构造二分图 $G=\langle V_1, V_2, E\rangle$,其中 $V_1=\{x_1, x_2, \cdots, x_8\}$, $V_2=\{y_1, y_2, \cdots, y_8\}$ 分别对应于棋盘的 8 行 8 列,当且仅当属于行 x_i 且属于列 y_j 的方格是被选出时,边 $\{x_i, y_j\}$ 属于 $E(G)$,于是,我们的图的边对应于所选的方格. 所以 G 是 2 正则二分图。而它有两个边不交的完美匹配,设为 M_1 与 M_2. 若把白的与黑的棋子分别放置在对应于 M_1 与 M_2 的边所对应的方格上,即得所求的放置法.

24 **证明** 在 V_1 中任取一个非空子集 S,记

$$E_0=\{\{x, y\} \mid x\in V_1, y\in V_2\}$$

则 $|E_0|=\sum_{x\in S}d_G(x)$

因此，S 中顶点度的平均数是 $|E_0|/|S|$．而 $N_G(S)$ 中顶点度的平均数至少是 $|E_0|/|N_G(S)|$．由题意

$$|E_0|/|S| \geqslant |E_0|/|N_G(S)|$$

因此，$|N_G(s)| \geqslant |s|$．即 G 中含有饱和 V_1 中所有顶点的匹配．

25　**证明**　设 M 是 G 的最大匹配，但 $|M| \leqslant k-1$．令 Q 表示 M 的边的端点集合，由于在 Q 内的顶点数为 $2|M| \leqslant 2k-2$，在 $V(G)-Q$ 中至少存在两个顶点，设为 x, y．根据 M 的最大性，$V(G)-Q$ 中任两个顶点不相邻。若存在 M 的一条边，使其中的一个端点相邻于 x，而另一个端点相邻于 y．则 G 存在 M 可增广路，这将矛盾于 M 是最大匹配．现在我们考虑 x 的所有邻点 $x_1, x_2, \cdots, x_l$，$l \geqslant k$．这 l 个点全在 Q 内，让我们分别用 $y_1, y_2, \cdots, y_l$ 表示在 M 下与 $x_1, x_2, \cdots, x_l$ 配对的顶点，则 y 不能与 $y_1, y_2, \cdots, y_l$ 相邻．由此可得

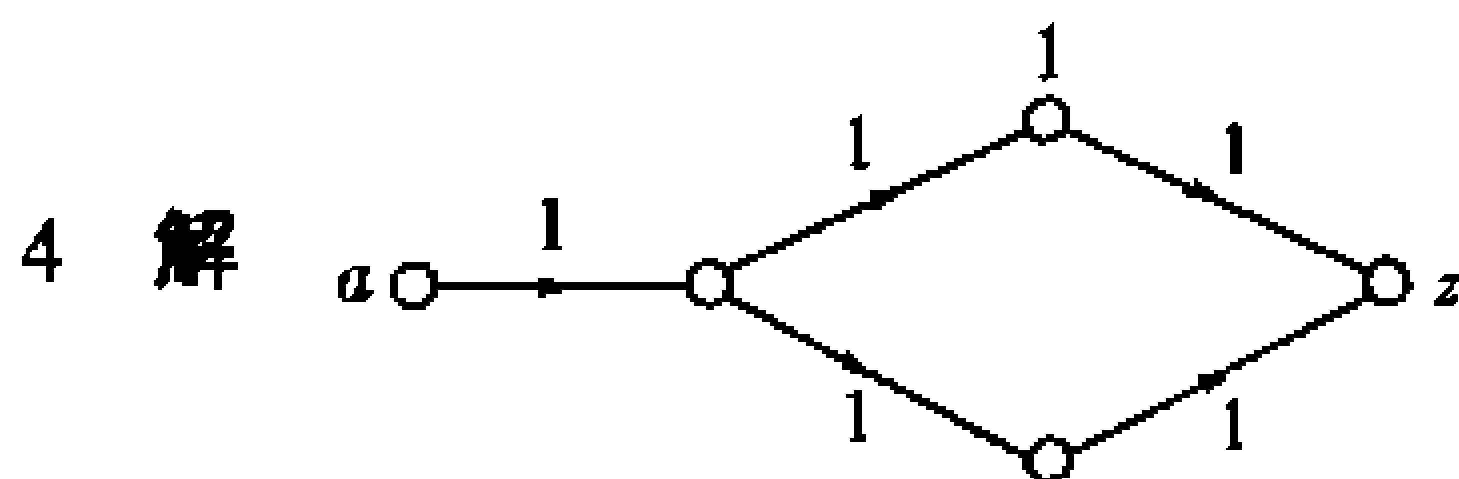
$$d_G(y) \leqslant (2k-2)-l \leqslant k-2$$

与 G 的假设矛盾．

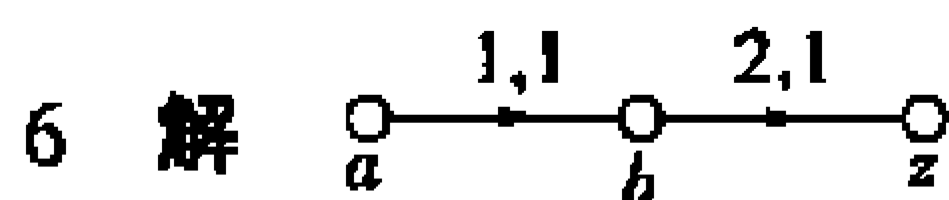
习　题　六

1　**解**　(a) $\langle b, c\rangle$ 是 6,3；$\langle a, d\rangle$ 是 4,2；$\langle c, e\rangle$ 是 6,1；$\langle c, z\rangle$ 是 5,2．流的值是 5．

4　**解**



6　**解**



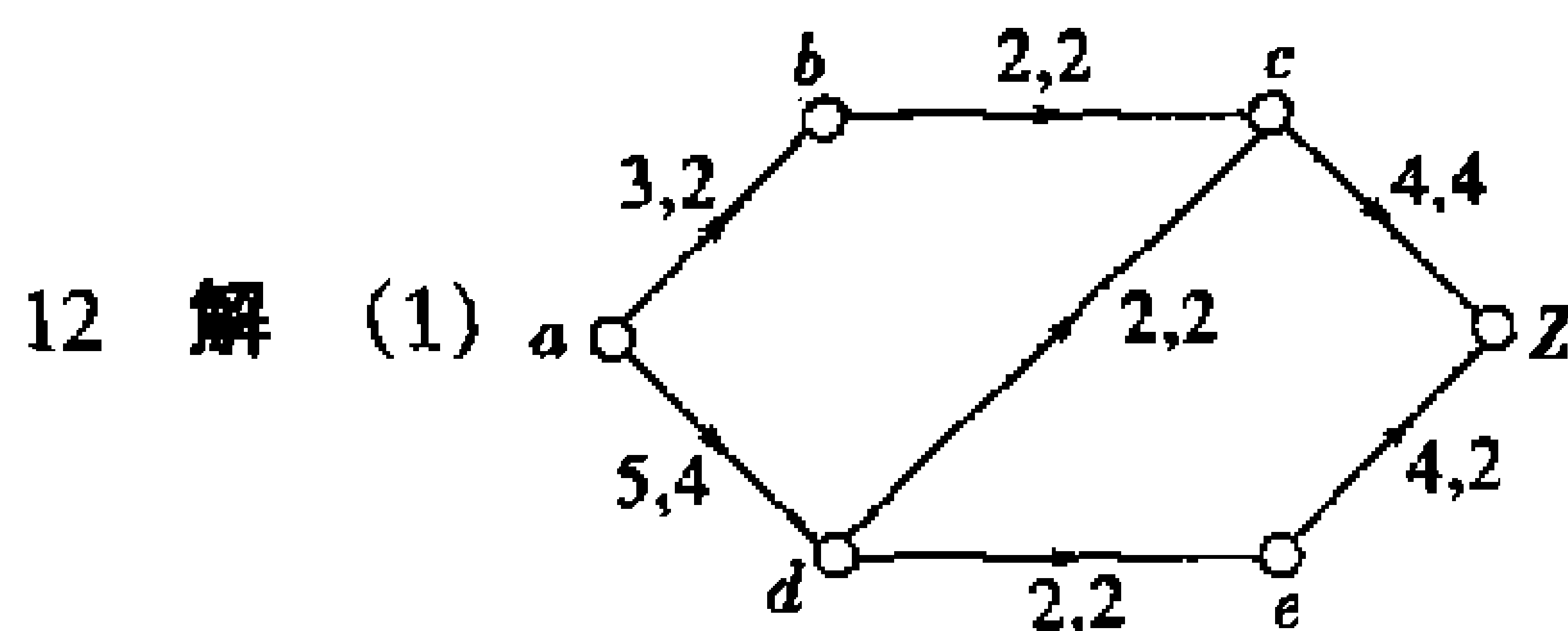
$$C_{ab}=1, C_{bc}=2, m_{ab}=1, m_{bz}=2$$

9 **解** 考虑流 $a \xrightarrow{1,1} b \xrightarrow{2,1} z$ 和切割 $p=\{a,b\}$.

10 **解** (1)1

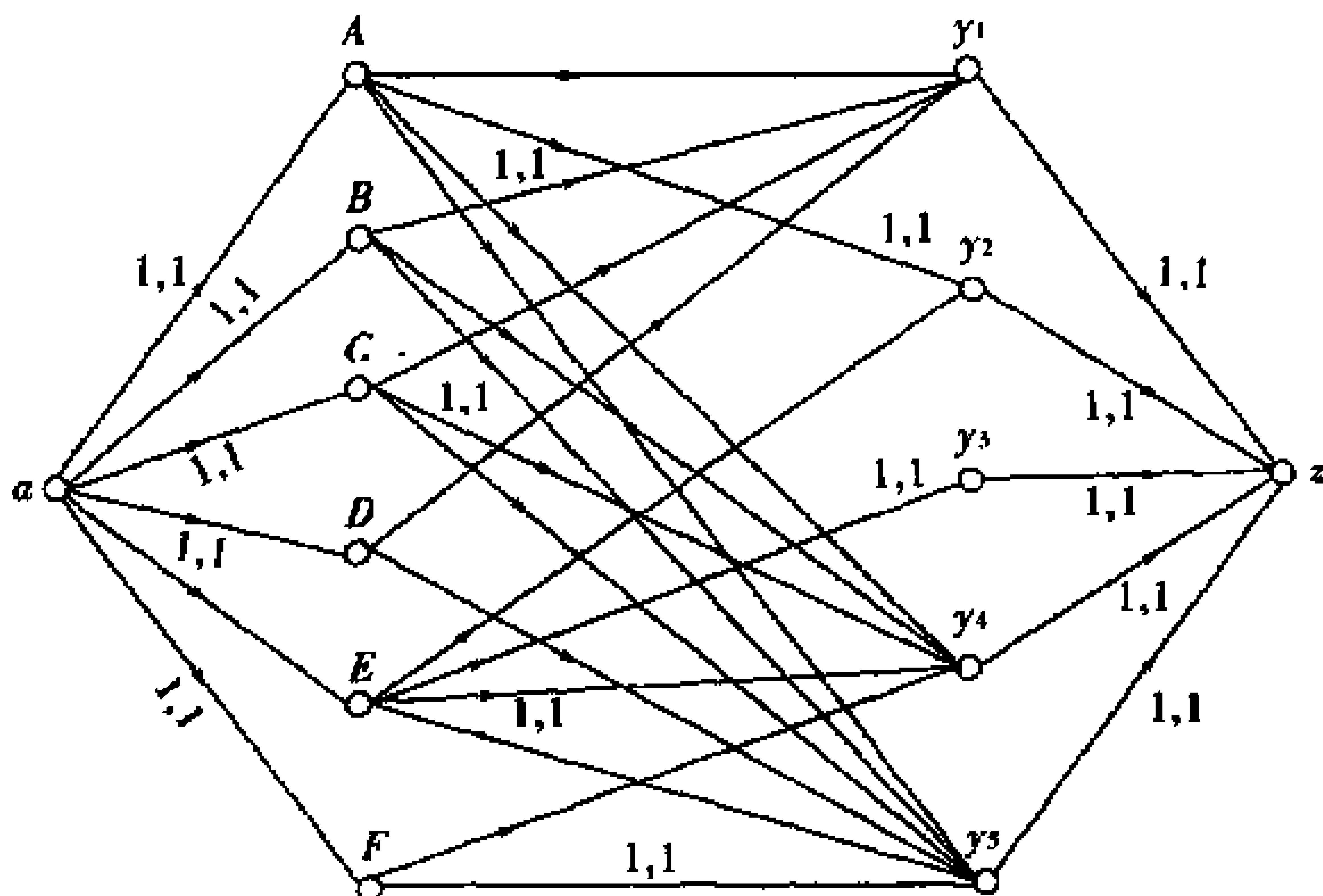
11 **解** (1) $F_{ab}=6, F_{ac}=0, F_{ad}=3, F_{be}=6, F_{ce}=0, F_{df}=3$ $F_{eh}=4, F_{eg}=2, F_{fg}=3, F_{gh}=2, F_{gp}=3, F_{hz}=6, F_{pz}=3$.

12 **解** (1)

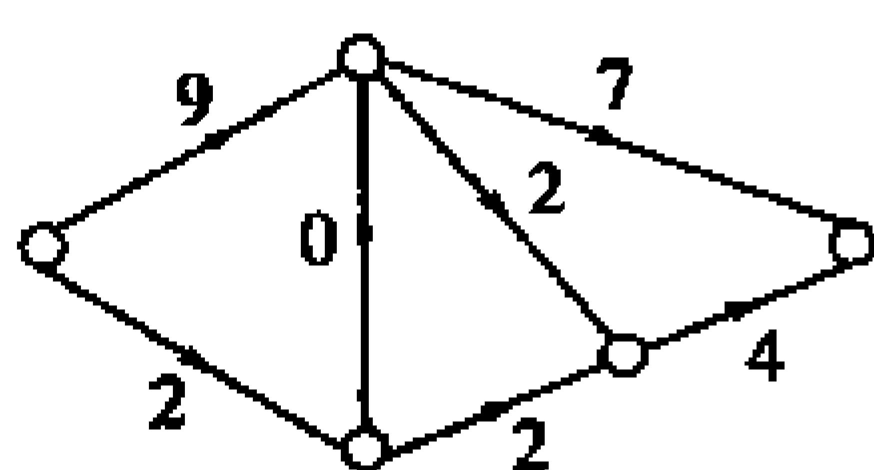


(2) 19

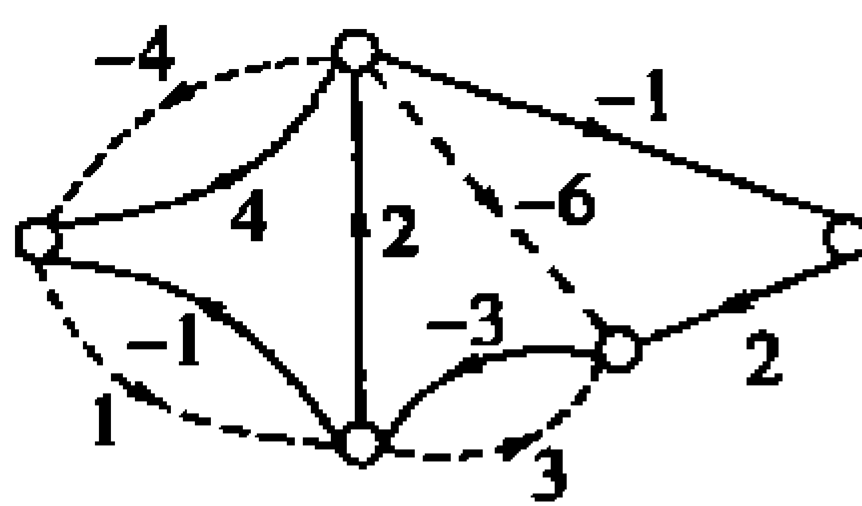
19 **解** 所有没标记的边是 1,0.不存在完美匹配.



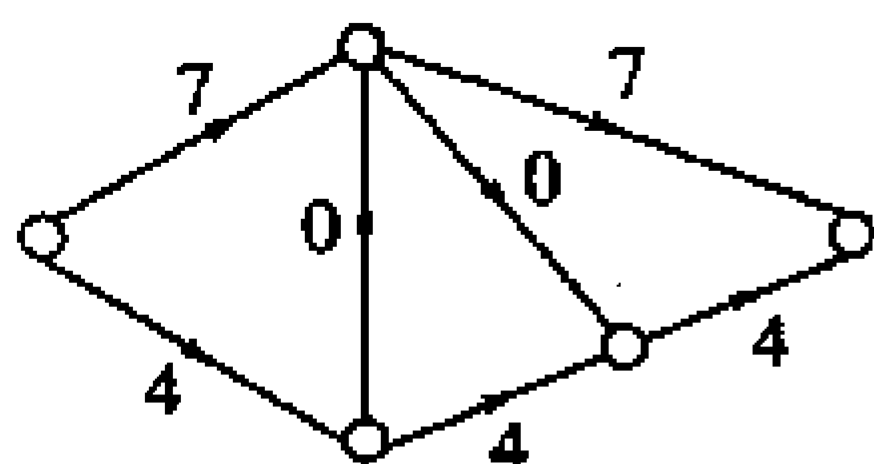
20 解



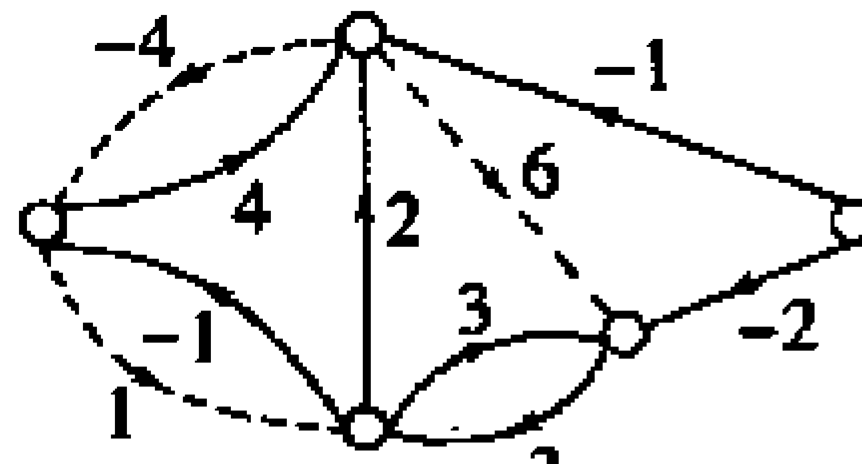
(1)



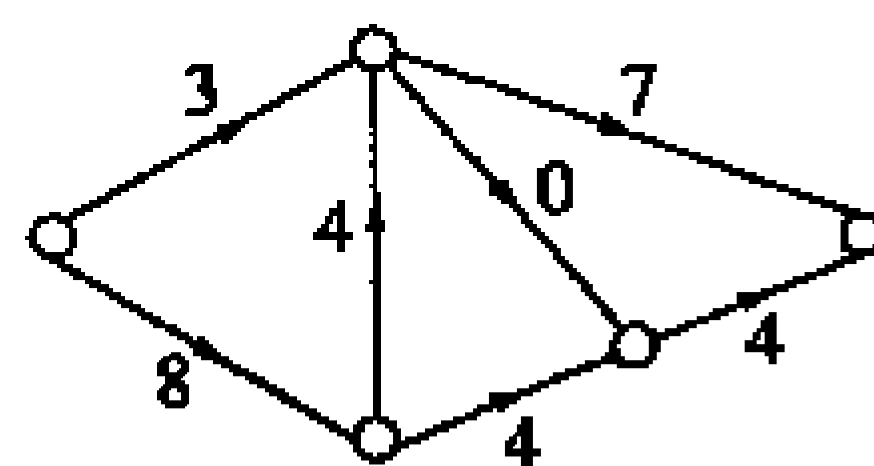
(2)



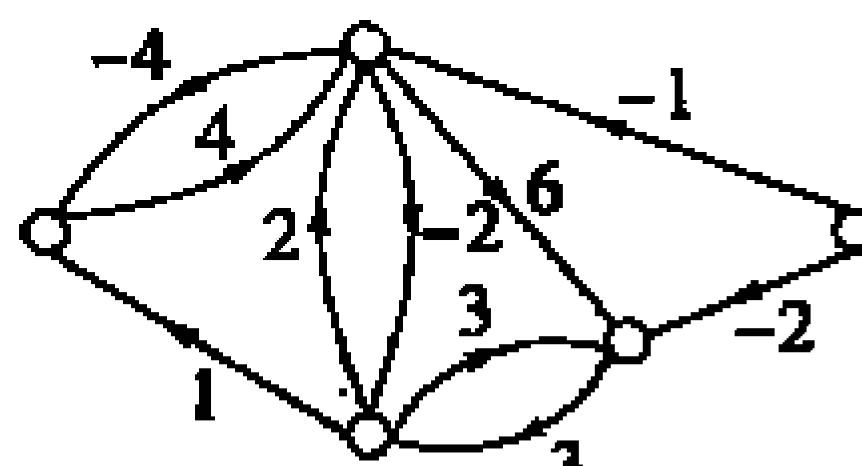
(3)



(4)



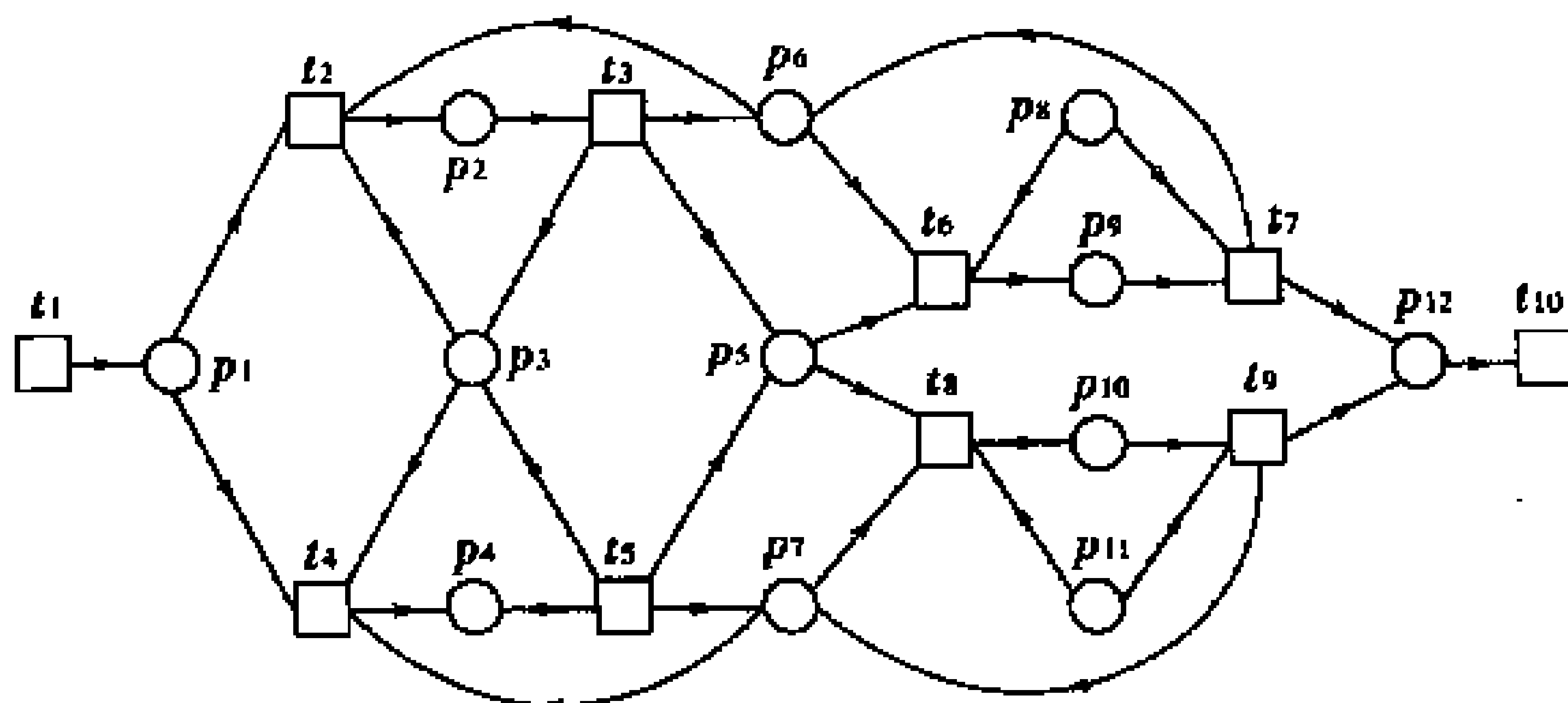
(5)



(6)

虚线表示的是负圈.

21 解



其中　t_1:任务等待加工;

t_2:A 操作 m_1;

t_3:操作 m_1 完毕;

t_4:B 操作 m_1;

t_5:B 操作 m_1 完毕;

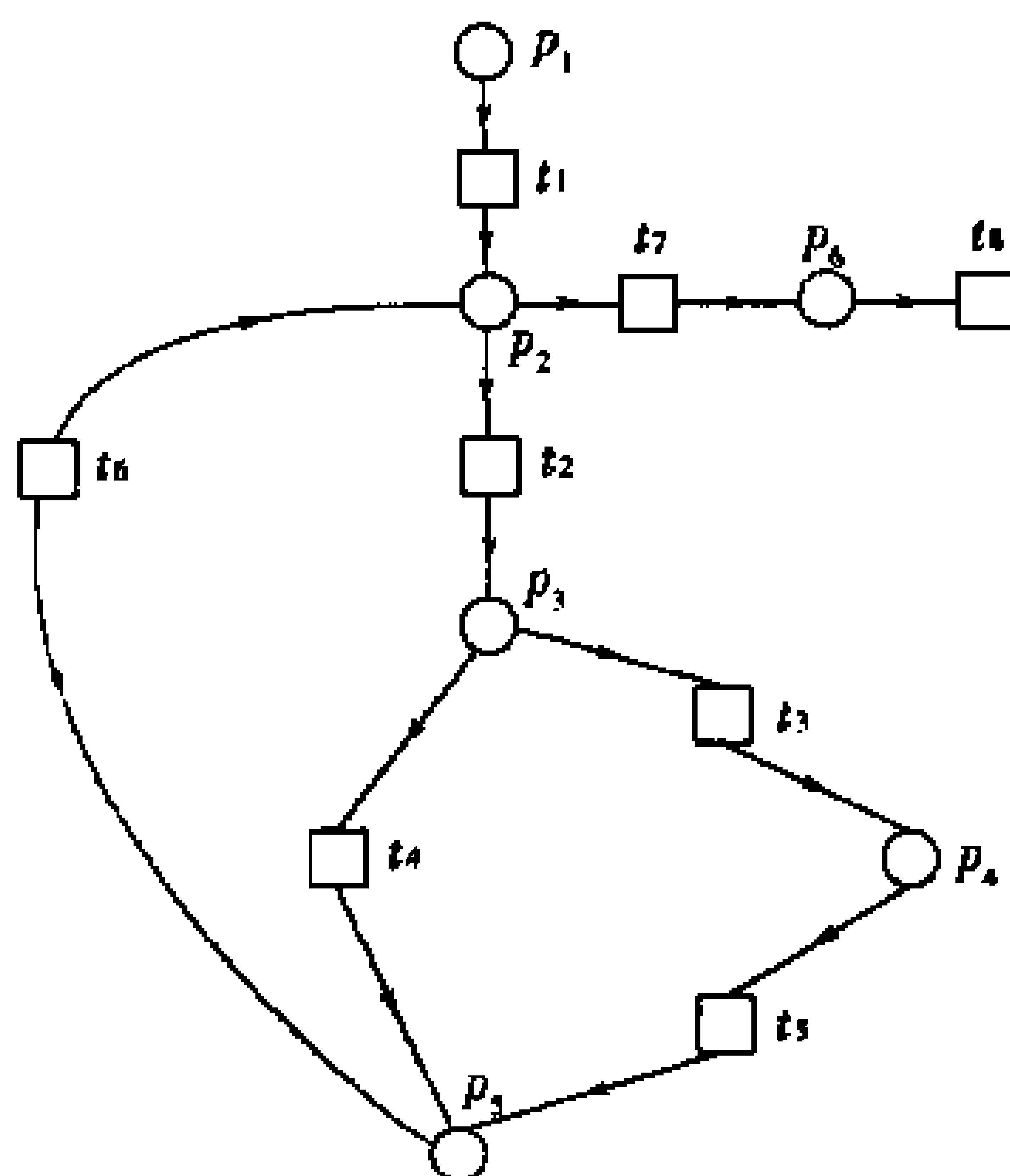
t_6:A 操作 m_2;

t_7:A 结束 m_2 的操作;

t_8:B 操作 m_3;

t_9:B 结束 m_3 的操作;

t_{10}:将加工完毕的任务送走.



22 解

其中 t_1:完成 S_1 运算;

t_2:$y_1>0$;

t_3:y_1 是奇数;

t_4:y_1 是偶数;

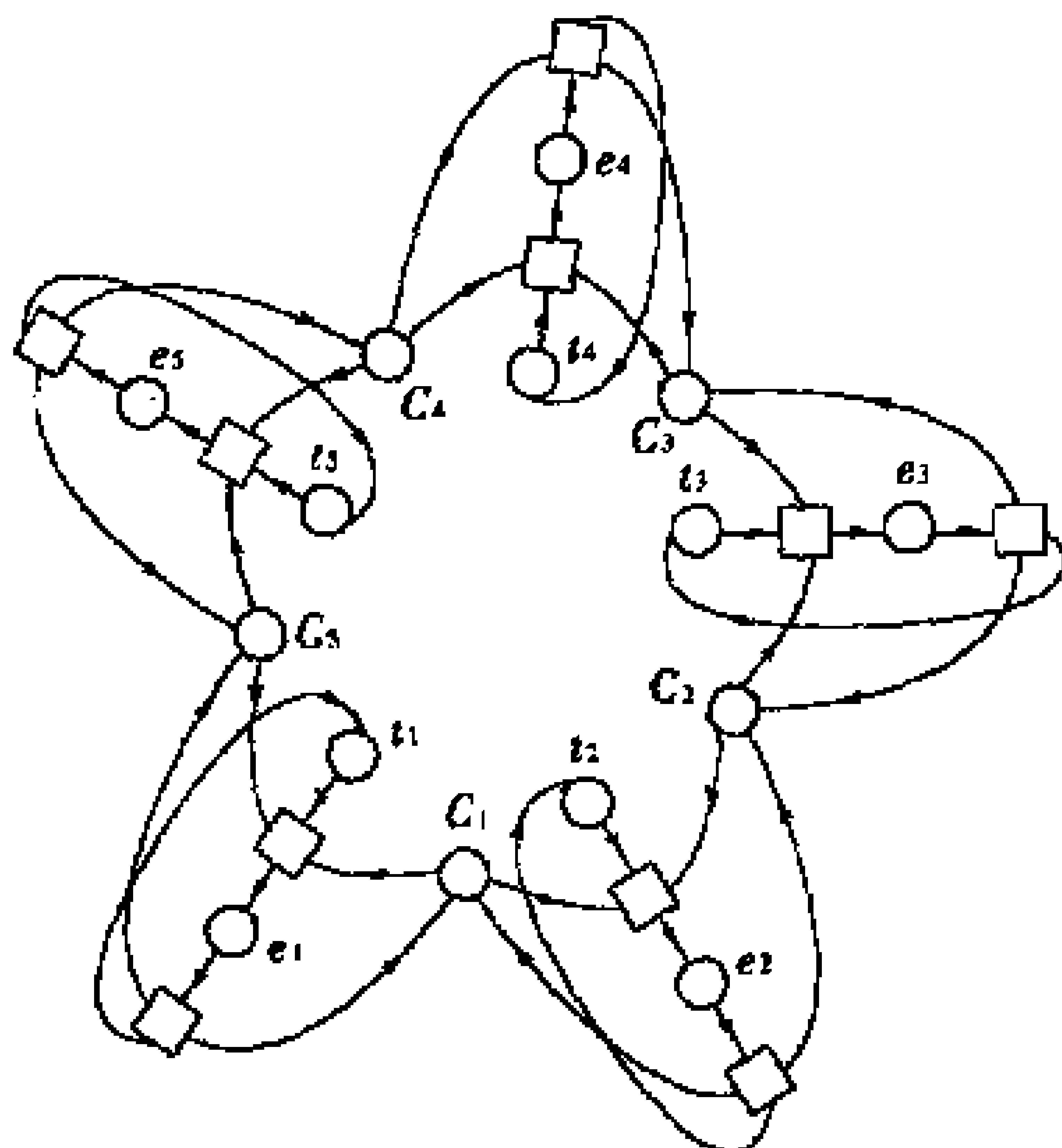
t_5:完成 $y_3 \leftarrow y_3 * y_2, y_1 \leftarrow y_1 - 1$;

t_6:$y_2 \leftarrow y_2^2, y_1 \leftarrow y_1 - 2$;

$t_7: y_1 \leqslant 0$

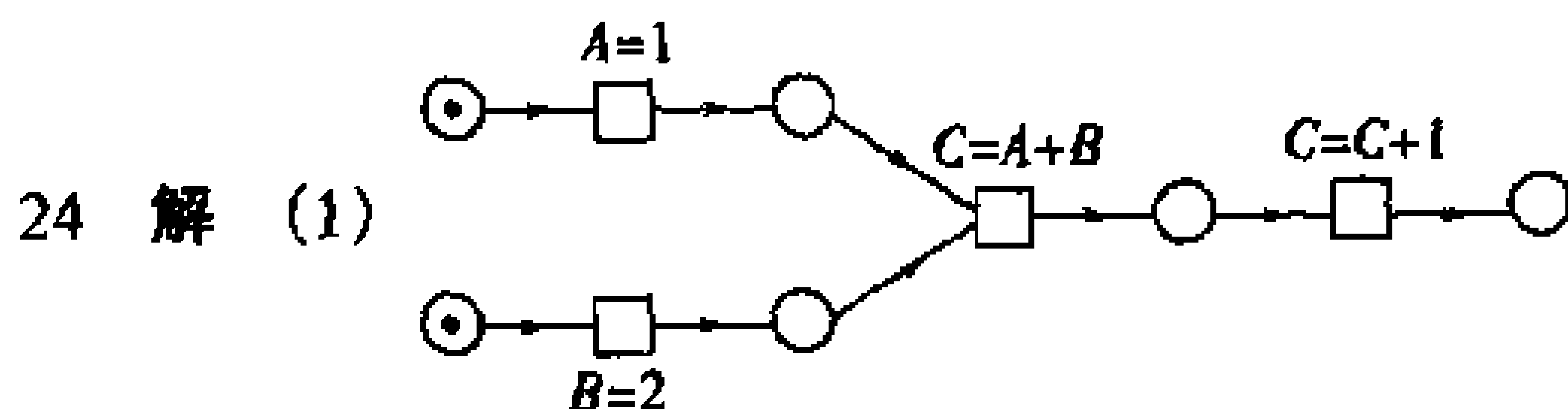
t_8:输出 y_3.

23 解



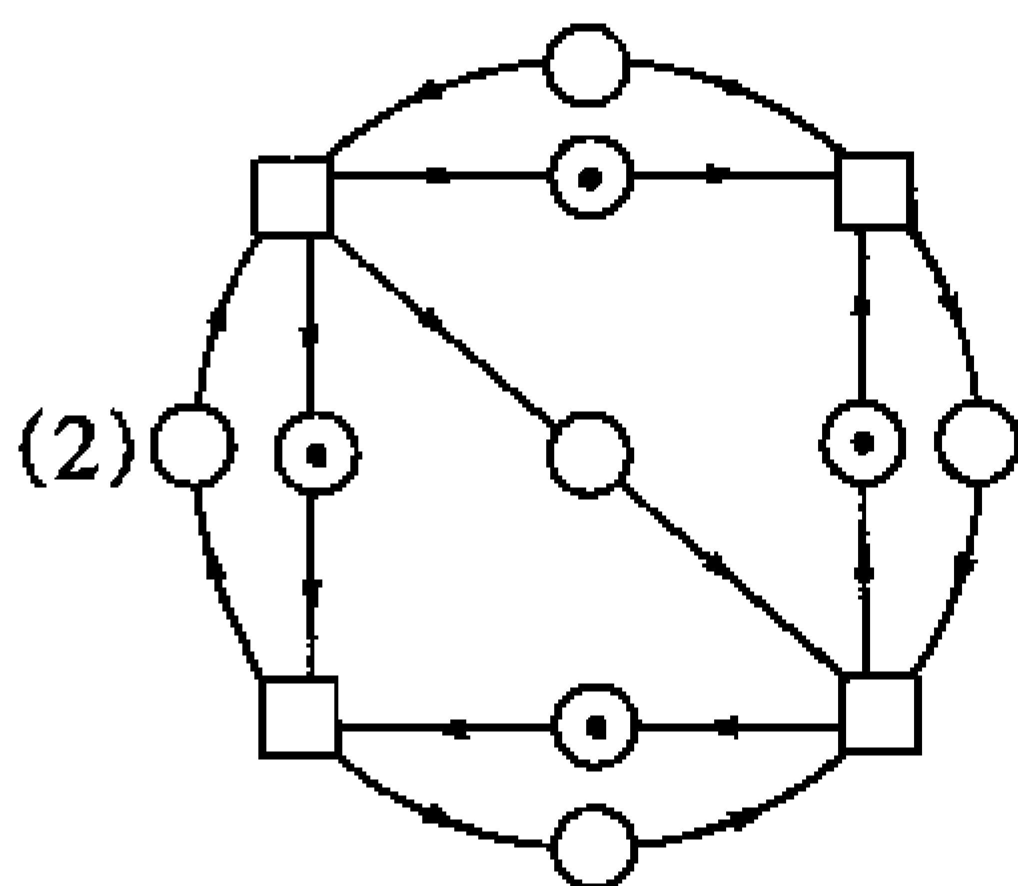
其中 c_1, c_2, c_3, c_4, c_5 为 5 根筷子,e_i 和 t_i 分别表示第 i 个哲学家吃饭和想问题两种状态,$i = 1, 2, \cdots, 5$,当哲学家从思索状态转入吃的状态,则他必须有左右两根筷子.

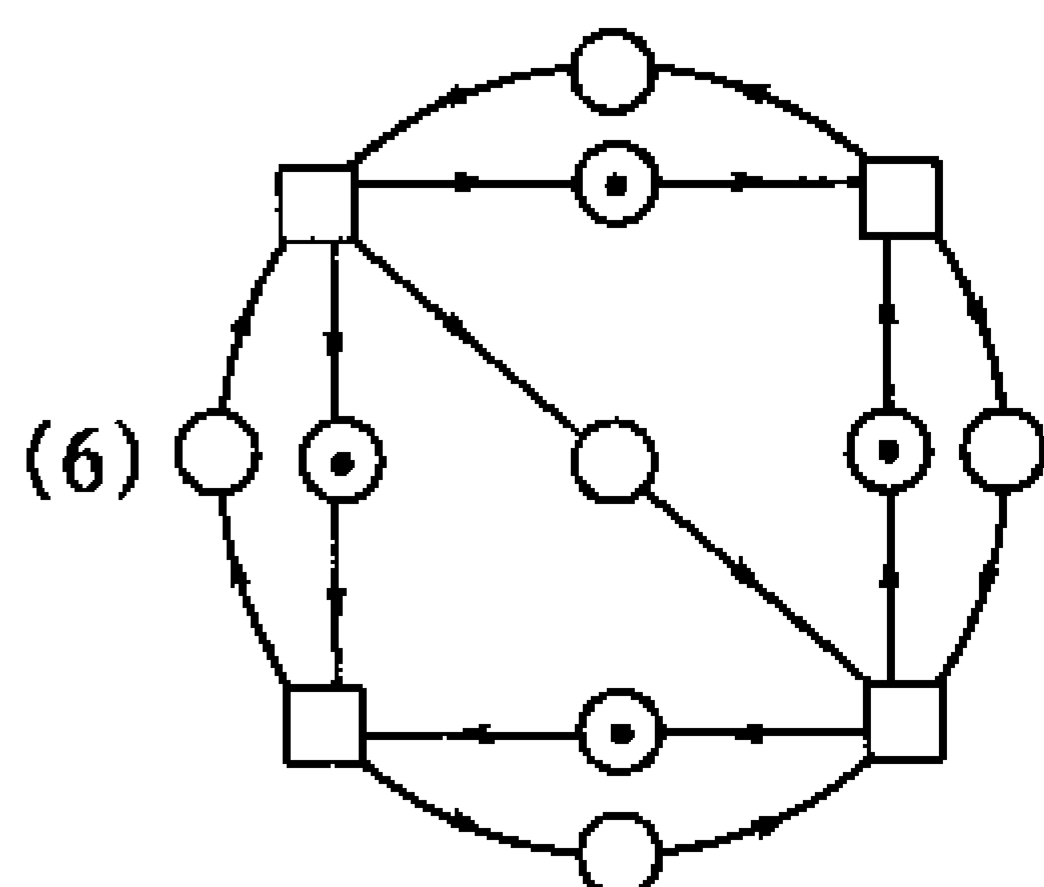
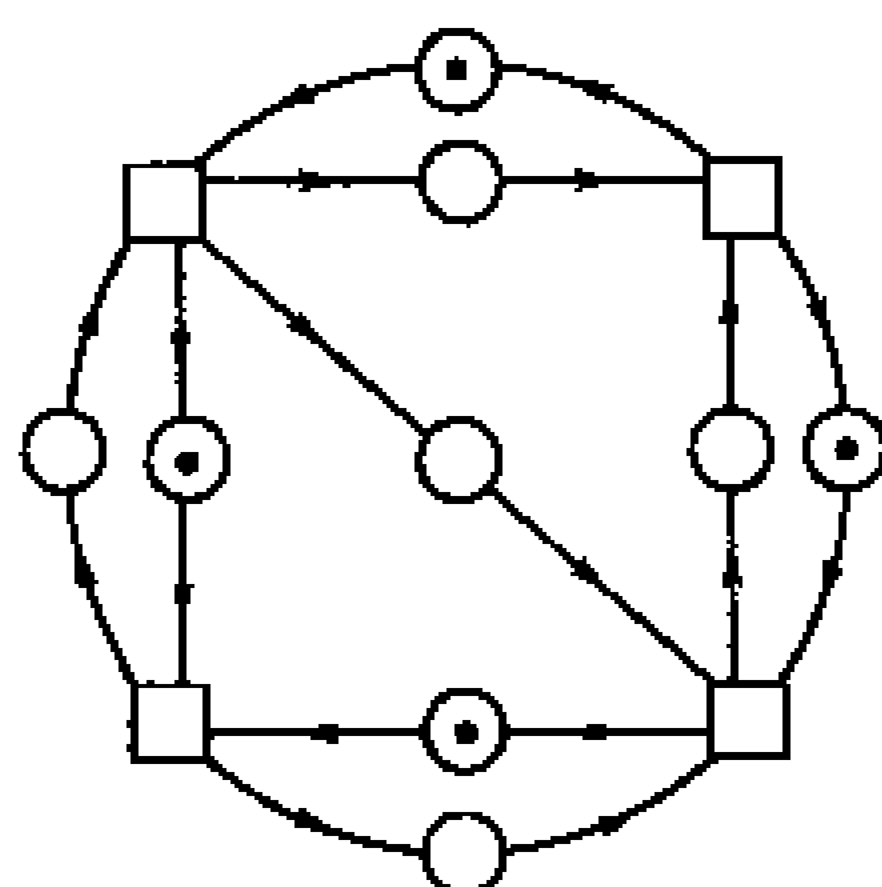
24 解 (1)

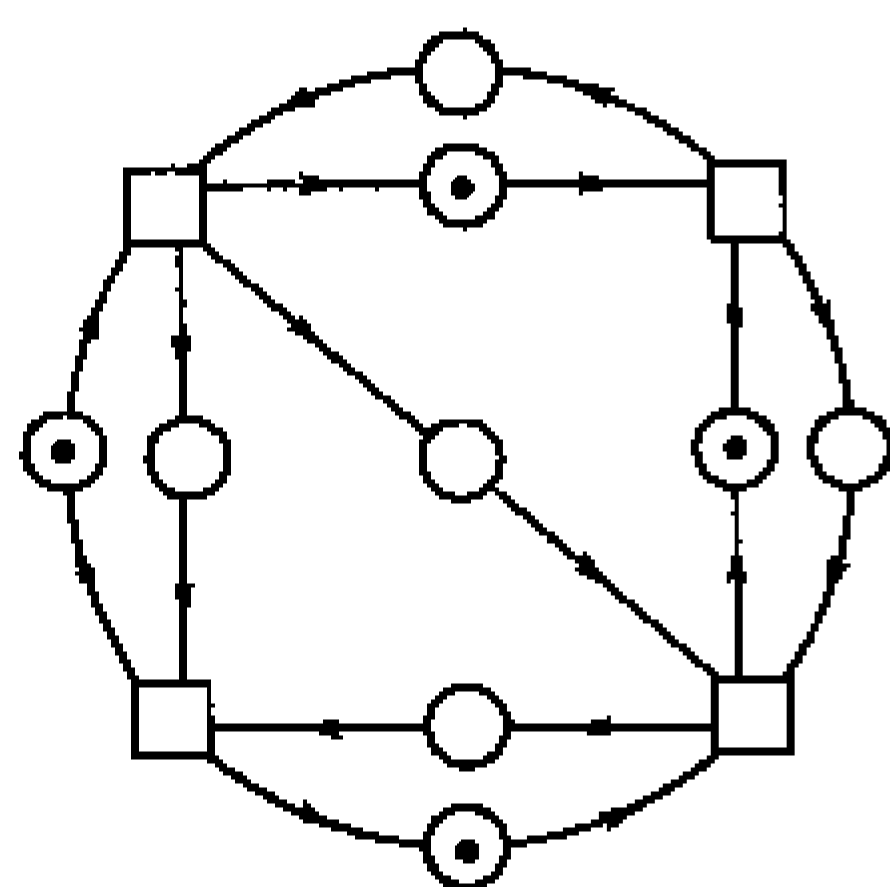
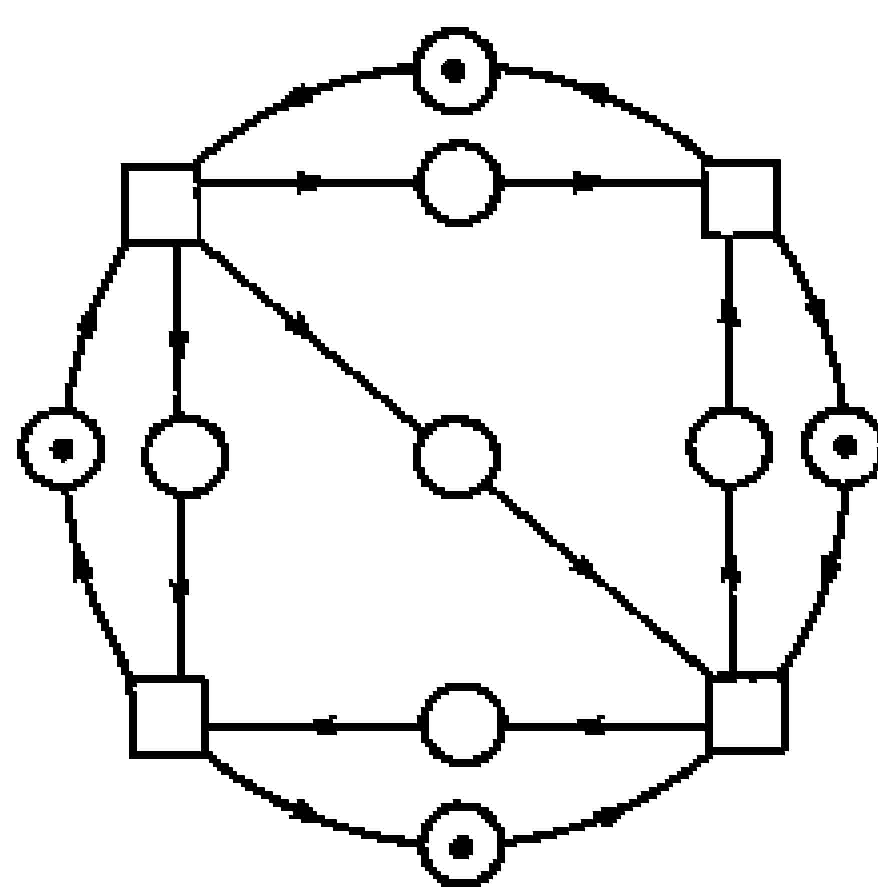


25 解 (a)

(1)t_1　　(3)是　　(4)是　　(5)是

(2) 

(6)  

(7)任何标记至少有一个位置放置两个标记.

参考文献

[1] 卜月华,吴建专,顾国华,殷翔.图论及其应用.东南大学出版社,2002.
[2] 蒋长浩.图论与网络流.中国林业出版社,2001.
[3] 徐俊明.图论及其应用.中国科学技术大学出版社,2000.
[4] 王朝瑞.图论.北京理工大学出版社,2000.
[5] 卢开澄,卢华明.图论及其应用(第二版).清华大学出版社,1996.
[6] 王树禾.图论及其算法.中国科学技术大学出版社,1994.
[7] 陈子岐,朱必文,刘峙山.图论,高等教育出版社,1990.
[8] 田丰,马仲蕃.图与网络流理论.科学出版社,1987.
[9] 宋增民.图论与网络最优化.东南大学出版社,1990.
[10] (美)Kenneth H. Rosen. *Discrete Mathematics and Its Applications*,英文版,第4版.机械工业出版社,1999
[11] Bernard Kolman, Robert C. Busby, Sharon Cutler Ross. *Discrete Mathematical Structures*,英文版,第4版.高等教育出版社,2001.
[12] Bondy, J. A. and Murty, U. S. R.. *Graph theory with applications*. The Maclmillan Press Ltd, 1976,中译本:J.A.邦迪,U.S.R.默蒂著.图论及其应用.科学出版社,1984.
[13] Bela Bollobas. *Modern Graph Theory*. Springer, New York 1998.
[14] Lawler, E. L. *Combinatotial optimization: networks and matroids*. Holt Rienhart and Winston, 1986.
[15] Richard Johnson baugh(著),王孝喜,邵秀丽,朱思俞(译);离散数学(第4版).电子工业出版社,1999.
[16] (美)Knneth H. Rosen(著),袁崇义,屈婉玲,王捍贫,刘田(译).离散数学及其应用(原书第4版).机械工业出版社,2002.
[17] 耿素云.离散数学习题集(图论分册).北京大学出版社,1998.
[18] 张克民,林国宁,张忠辅.图论及其应用习题解答.清华大学出版社,1989.

[19] 左孝凌,李为鑑,刘永才.离散数学——理论·分析·题解.上海科学技术文献出版社,1998.
[20] 左孝凌,李为鑑,刘永才.离散数学,上海科学技术文献出版社,1998.
[21] 朱一清.离散数学.电子工业出版社,1998.
[22] 方世昌.离散数学.西安电子科技大学出版社,2001.
[23] 耿素云,屈婉玲,张立昂.离散数学(第二版).清华大学出版社,2002.
[24] 陈莉,刘晓霞.离散数学.高等教育出版社,2002.
[25] 杨振生(编著),王树禾(审).组合数学及其算法,中国科学技术大学出版社,1997.
[26] 李兢,刘长林,申石虎.关于图的极大独立集的理论及生成算法.电子学报:(8),1995.
[27] 张大方.基于矩阵的极大独立点集生成算法.电子学报:(5),1998.



ISBN 7-312-01558-1/O·270

定价：18.00元